ARPA ORDER NO. 189-61

MEMORANDUM
RM-4549-ARPA
AUGUST 1965

COMPUTATIONAL ASPECTS OF INVERSE PROBLEMS IN ANALYTICAL MECHANICS, TRANSPORT THEORY, AND WAVE PROPAGATION

Harriet Kagiwada

PREPARED FOR:

ADVANCED RESEARCH PROJECTS AGENCY

The RAND Corporation
SANTA MONICA • CALIFORNIA

DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST QUALITY AVAILABLE.

COPY FURNISHED CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

MISSING PAGE NUMBERS ARE BLANK AND WERE NOT FILMED

ARPA ORDER NO. 189-31

MEMORANDUM
RM-4549-ARPA
AUGUST 1965

COMPUTATIONAL ASPECTS OF INVERSE PROBLEMS IN ANALYTICAL MECHANICS, TRANSPORT THEORY, AND WAVE PROPAGATION

Harriet Kagiwada

This research is supported by the Advanced Research Projects Agency under Contract No. SD-79. Any views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of ARPA.

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

PREFACE

Inverse problems are basic problems in science, in which physical systems are to be identified on the basis of experimental observations. It is shown in this Memorandum that a wide class of inverse problems may be readily solved with high speed computers and modern computational techniques. This is demonstrated by formulating and solving some inverse problems which arise in celestial mechanics, transport theory and wave propagation. FORTRAN programs are listed in the Appendix. Computational aspects of inverse problems are of interest to physicists, engineers and biologists who are engaged in system identification, in the planning of experiments and the analysis of data, and in the construction of mathematical models. This study was supported by the Advanced Research Projects Agency.

The author wishes to express her gratitude for the inspiration and guidance of Professor Sueo Ueno of the University of Kyoto, and of Dr. Richard Bellman and Dr. Robert Kalaba of the RAND Corporation.

SUMMARY

Inverse problems are basic problems in science, in which physical systems are to be identified on the basis of experimental observations. Inverse problems are especially important in the fields of astrophysics and astronomy, for their objects of investigation are frequently not observable in a direct fashion. Solar and stellar structure, for example, is estimated from the study of spectra, while the structure of a planetary atmosphere may be deduced from measurements of reflected sunlight.

We show that a wide class of inverse problems may now be solved with high speed computers and modern computational techniques. Many problems may be formulated in terms of systems of ordinary differential equations of the form

(1) $$\dot{x} = f(x, \alpha) .$$

Here, t is the independent variable, x is an n-dimensional vector whose components are the dependent variables, and α is an m-dimensional vector whose components represent the structure of the system. For instance, in an orbit determination problem, Eqs. (1) are the dynamical equations of motion, and the masses of the bodies involved may be given by the vector α. When the system parameters and a complete set of initial conditions,

(2) $$x(0) = c ,$$

are known, an integration of (1) produces the solution $x(t)$ on the interval $0 \leq t \leq T$. This is done speedily and accurately with a digital computer.

On the other hand, in an inverse problem, the solution $x(t)$ or some function of $x(t)$ is known at various times, while the parameters are not directly observable. We wish to determine the structure of the system as given by the parameter vector, α, and a complete set of initial conditions, c. We regard this as being a nonlinear boundary value problem in which the unknowns are some of the c's and α's. We require that the solution agree with the observations,

$$x(t_i) \cong b_i \quad (3)$$

in some sense, e.g., in a least squares sense.

Frequently, problems which do not naturally occur in the form of systems of ordinary differential equations may be expressed in that form in an approximate representation. In this thesis, we show how we may reduce a partial differential-integral equation to a system of ordinary differential equations with the use of a quadrature formula. Also, we may express a partial differential equation, like the wave equation,

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2}, \quad (4)$$

in the desired form by applying Laplace transform methods which remove the time derivative. Other possibilities are clearly available.

Nonlinear boundary value problems can be solved by a variety of methods, which include quasilinearization, dynamic programming, and invariant imbedding. These techniques are especially suited to modern computers, for they reduce nonlinear boundary value problems to nonlinear initial value problems, which are more easily treated on digital computers.

These computational ideas are illustrated in this thesis by actually formulating and solving some inverse problems which arise in celestial mechanics, radiative transfer, neutron transport and wave propagation. In one of the problems, we estimate the stratification of a layered medium from reflection data. In another, we determine a variable wave velocity by observing a portion of the transients produced by a known stimulus. Numerical experiments are conducted to estimate the stability of the methods and the effect of the number and quality of observational measurements. Complete FORTRAN programs are given in the Appendices.

These computational aspects of inverse problems may prove to be of value to the physicist, engineer, or mathematical biologist who wishes to determine the structure of a system on the basis of observations. These ideas may be helpful in the planning of experiments and in the choice of apparatus. They may be used to design systems which have certain desired properties. In particular, these methods may be useful in the construction of stellar and planetary models.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

CHAPTER ONE

INTRODUCTION

1. INTRODUCTION

Inverse problems are fundamental problems of science [1-12]. Man has always sought knowledge of a physical system beyond that which is directly observable. Even today, we try to understand the dynamical processes of the deep interior of the sun by observing the radiation emerging from the sun's surface. We deduce the potential field of an atom from nuclear scattering experiments. The underlying theme is the relationship between the internal structure of a system and the observed output. The hidden features of the system are to be extracted from the experimental data.

Mathematical treatment of physical problems has been devoted almost exclusively to the "direct problem." A complete picture of the system is assumed to be given, and equations are derived which describe the output as a function of the system parameters. The inverse problem is to determine the parameters and structure of a system as a function of the observed output.

One can solve a given inverse problem by solving a series of direct problems: by assuming different sets of parameters, determining the corresponding outputs from the theoretical equations, and comparing theoretical versus experimental results. By trial and error, one may find a solution which approximately agrees with the experimental data. This is not a very efficient procedure. Another way to solve an inverse problem is to solve analytically for the unknown parameters as functions of the measurements. This method generally requires much abstract mathematics and simple approximations of complex functions. The resultant inverse solution may be valid only in very special circumstances.

What we seek are efficient, systematic procedures for solving a wide class of inverse problems - procedures which are suitable for execution on high speed digital computers. Computers are currently capable of integrating large systems of ordinary differential equations, given a complete set of initial conditions, with high accuracy. We would like to formulate our problems in terms of systems of ordinary differential equations. Partial differential equations, such as the wave equation,

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u(x,t)}{\partial t^2} ,$$

may be reduced to systems of ordinary differential equations in several ways which include the use of Laplace transform methods, Fourier decomposition, and finite difference schemes. Integro-differential equations, which frequently occur in transport theory, may be reduced to systems of ordinary differential equations by approximating the definite integrals by finite sums using Gaussian and other quadrature formulas. Other means of formulating problems in terms of ordinary differential equations are possible.

We desire to formulate our inverse problem in such a way that we deal with ordinary differential equations. First, as we shall show, we may express the problem as a nonlinear boundary value problem, in which we seek a complete set of initial conditions. The unknown system parameters will be calculated directly from the initial conditions. Next, we resolve the nonlinear boundary value problem, ordinarily a difficult task, by the use of some sophisticated techniques [13-24]. We may replace the nonlinear boundary value problem by a rapidly converging sequence of linear boundary value problems via the technique of quasilinearization [1-3,16,17]. We may, alternatively, treat the problem as a multi-stage decision process with the use of dynamic programming [18]. Or, we may solve directly for the missing initial conditions by applying the concept of invariant imbedding [13,24]. From the solution of the nonlinear boundary value problem, we immediately obtain knowledge of the internal structure of the system.

In this thesis, we discuss some of these relatively new concepts, computational techniques, and applications. Our examples from celestial mechanics, transport theory and wave propagation are physically motivated. No specialized background is required on the part of the reader beyond a knowledge of elementary physics. We intend to be self-contained in the mathematical derivations, except for those matters which are well-treated elsewhere, such as dynamic programming, linear programming, and the numerical inversion of Laplace transforms. Again, no special mathematical knowledge is needed beyond the level of ordinary differential equations and linear algebraic equations. We will, however, assume that we have at our disposal a high-speed digital computer with a memory of about 32,000 words, plus a library of computer routines for numerical integration, matrix inversion, and linear programming. Our basic assumption is that our computer can integrate large systems of ordinary differential equations rapidly and accurately [25,26].

In the first chapter, we wish to emphasize some important ideas. We are given geocentric observations of a heavenly body, taken at various times [27-29]. The orbit of this body lies in the potential field created by the sun and an unknown perturbing mass. We show how the mass may be identified and the orbital elements found. For simplicity, we assume that the position of the perturbing mass is given; if desired, the position as a function of time could also be estimated. Since we are virtually forced by our modern computers to take a fresh look at old problems, we are not concerned with conic sections. A new methodology, based on high speed digital computers, is developed. The technique of quasilinearization, described in this chapter, enables us to solve this inverse problem with a minimum of effort. In spite of the newness of this solution of a long-standing problem in celestial mechanics, we employ this example for purely illustrative purposes.

Transport theory is intimately concerned with the determination of radiation fields within scattering and absorbing media [30-38]. Our first problem in radiative transfer (Chapter Two) serves to exemplify the philosophy and application of invariant imbedding. We derive the basic integro-differential equation for the diffuse reflection function, and we reduce it to a system of ordinary differential equations by the method of Gaussian quadrature. Then we formulate an inverse problem for the determination of layers in a medium from knowledge of the diffusely reflected light. We outline the computational procedure, and we present our results. In Chapter Three, our setting is again an inhomogeneous scattering medium. We investigate the effects of errors in our measurements, the number and quality of the observations, and the criterion function, on the estimates of the medium. Our criteria are either of least squares type, which leads to linear algebraic equations, or of minimax form, which is suitable for linear programming. We also consider a variation of the inverse problem, the contruction of a model atmosphere according to certain specifications. In Chapter Four, we consider an anisotropically scattering medium. The phase function is to be determined on the basis of measurements of diffusely reflected radiation in various directions.

An inverse problem in neutron transport (Chapter Five) is solved in a novel way. The dynamic programming approach leads to a determination of absorption coefficients in a rod, from measurements of internal fields. The calculation is done by an exact method, and is compared with a calculation based on an approximate theory. The approximate theory is accurate and less costly in computing time.

As we have already mentioned, the partial differential wave equation may be reduced to a system of ordinary differential equations by Laplace transform methods or by Fourier decompositions. In Chapter Six, we deal with ordinary differential equations for the Laplace transforms of the disturbances. In these equations, time appears only as a parameter. Our measurements of the disturbances at various times are converted to the corresponding transforms by means of Gaussian quadrature. We solve a nonlinear boundary value problem in order to determine the system parameters. The inverse Laplace transforms may be obtained by a numerical inversion technique [22].

In Chapter Seven, we use a decomposition of the form $u(x,t) = u(x)e^{-i\omega t}$, corresponding to a steady-state situation of wave propagation. We probe an inhomogeneous slab with waves of different frequencies and we "measure" the reflection coefficients. We wish to determine the index of refraction as a function of distance in the medium. Invariant imbedding leads to ordinary differential equations for the reflection coefficients, with known initial conditions. The unknown index of refraction in the equations and the observations of terminal values of the reflection coefficients make this a nonlinear boundary value problem. Quasilinearization is used to solve the problem, and computational results are presented.

The final chapter is a general discussion of inverse problems. Appendices of all the FORTRAN programs written for the computational experiments are included.

2. DETERMINATION OF POTENTIAL

Consider the motion of a particle (or a wave) in a potential field $V = V(x, y, z; k_1, k_2, \ldots, k_n)$ where we recognize the dependence on physical parameters k_1, k_2, ..., k_n. Suppose that these parameters are unknown, and that we have observations of the motion of the particle at various times. We wish to determine the potential function on the basis of these measurements.

Consider the following situation. A heavenly body H of mass m moves in the potential field created by the sun and a perturbing body P, whose masses are M and m_p, respectively, and $m \ll m_p \ll M$. All of the bodies concerned lie in the ecliptic plane. The potential energy varies inversely as the distance from the sun, r_s, and from the perturbing body, r_p,

$$(1) \qquad V = -\frac{k_s}{r_s} - \frac{k_p}{r_p} .$$

Here, k_s and k_p are the parameters

$$(2) \qquad k_s = \gamma m M, \; k_p = \gamma m m_p ,$$

where γ is the constant of gravitation. The quantity k_s may be assumed to be known. We choose our units so that $k_s = m$, or $\gamma M = 1$. The parameter k_p is unknown and $k_p < k_s$. We wish to determine k_p and thus V by observing the motion of H.

Let us take the plane of the ecliptic to be the (x, y) plane. The sun is situated at the origin, the earth at the point $(1, 0)$, and the perturbing body at the location $(\xi, \eta) = (4, 1)$. The earth only enters into the discussion as the point from which measurements are taken. Its mass is neglected. The potential function is

$$(3) \qquad V(x,y;\, k_p) = -\frac{k_s}{(x^2+y^2)^{1/2}} - \frac{k_p}{[(\xi-x)^2+(\eta-y)^2]^{1/2}} \; .$$

Angular observations of H are made at various times t_i, $i = 1, 2, \ldots, 5$. Fig. 1 illustrates the physical situation. Each solid arrow points to H at a given time t_i. The angle between the line of sight and the x axis is the observation. For comparison, see the dashed arrows which point to H when the mass of P is exactly zero, i.e., when $k_p = 0$. It is obvious that k_p is small.

The equations of motion are

$$(4) \qquad \begin{aligned} \ddot{x} &= \frac{-x}{(x^2+y^2)^{3/2}} + \frac{\alpha(\xi-x)}{[(\xi-x)^2+(\eta-y)^2]^{3/2}} \; , \\ \ddot{y} &= \frac{-y}{(x^2+y^2)^{3/2}} + \frac{\alpha(\eta-y)}{[(\xi-x)^2+(\eta-y)^2]^{3/2}} \end{aligned}$$

where the parameter α,

$$(5) \qquad \alpha = \frac{k_p}{k_s} = \frac{m_p}{M} \; ,$$

is the mass of P relative to the mass of the sun. At times t_i, we obtain the angular data $\theta(t_i)$ which are, in radians,

$$(6) \qquad \begin{aligned} \theta(0.0) &= 0.0 \; , \\ \theta(0.5) &= 0.252188 \; , \\ \theta(1.0) &= 0.507584 \; , \\ \theta(1.5) &= 0.763641 \; , \\ \theta(2.0) &= 1.01929 \; . \end{aligned}$$

We wish to determine α, $x(0)$, $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ so that the conditions

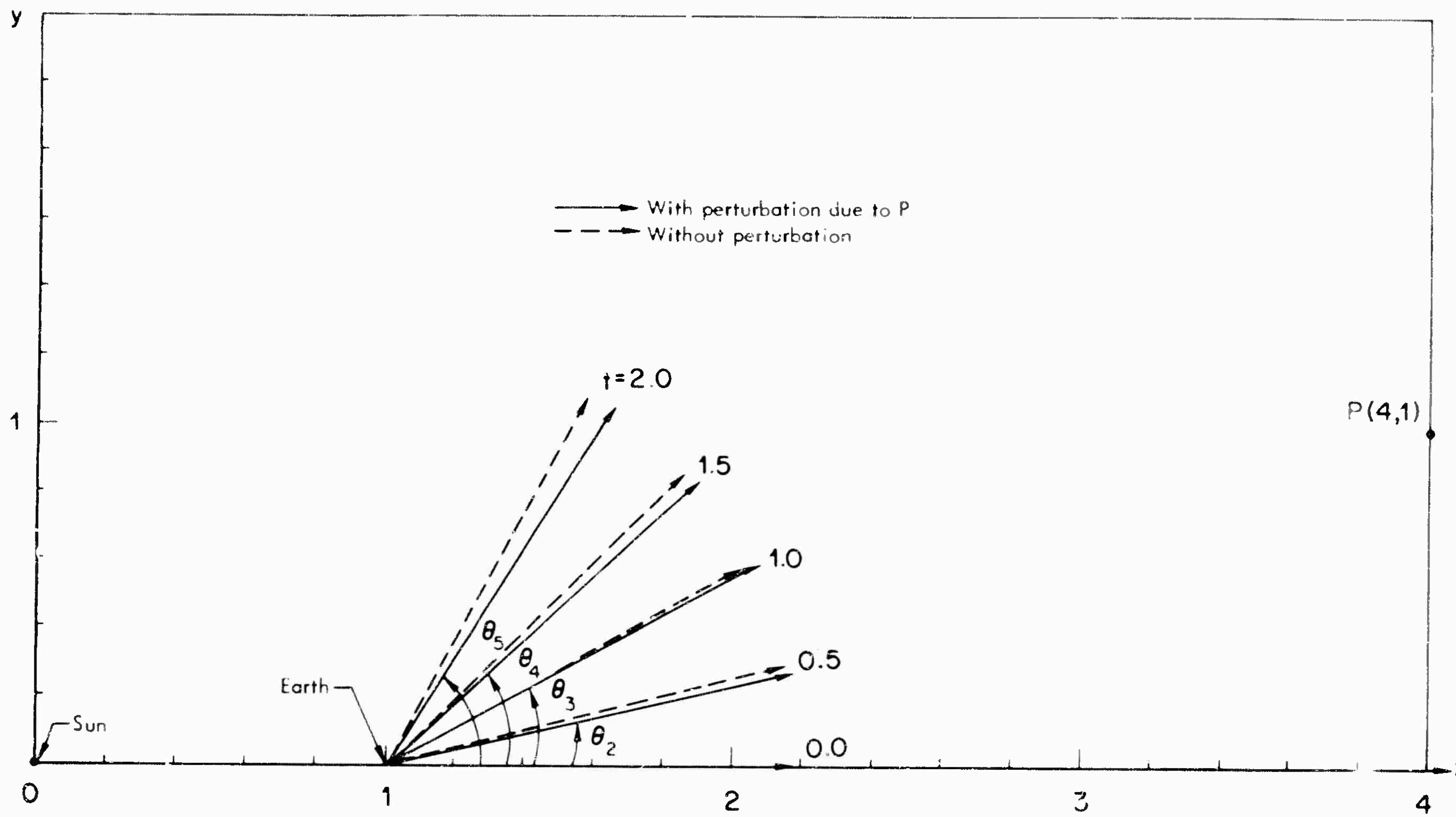


Fig. 1. Angular observations of a heavenly body.

(7) $$\tan \theta(t_i) = \frac{-y(t_i)}{1-x(t_i)}$$

are fulfilled. This is a nonlinear multipoint boundary value problem. The solution of this problem gives the relative mass of the perturbing body and the orbit of H as a function of time. The potential (3) is determined when α is known. We may consider the problem then to be the determination of the orbit [19,23,27-29].

For an arbitrary potential field, we are unable to express the solution analytically. We solve the problem computationally using the technique of quasilinearization [16,17].

3. QUASILINEARIZATION, SYSTEM IDENTIFICATION AND NONLINEAR BOUNDARY VALUE PROBLEMS

Consider a physical system or process which is described by the system of N equations

(1) $$\dot{x} = f(x, \alpha),$$

where x is a vector of dimension N, a function of independent variable t, with the N initial conditions

(2) $$x(0) = c.$$

The vector x describes the state of the system at "time" t, and α is a parameter vector of the system. With α given, Eqs. (1) and (2) completely describe the system, for the state at any time t, x(t), may be calculated by a numerical integration of (1) with initial conditions (2).

Now let us suppose that we have a system described by Eqs. (1), but α is unknown to us, and the initial conditions (2) are also unknown. However, we are able to make measurements of certain components of the state of the system at various times t_i. We wish to identify the system by determining α, and we wish to find a complete set of initial conditions $x(0) = c$ so that the system is fully described.

We think of the system parameter vector as if it were a dependent variable which satisfies the vector equation

$$\dot{\alpha} = 0 \tag{3}$$

with the unknown initial conditions

$$\alpha(0) = \alpha_0. \tag{4}$$

The multipoint boundary value problem which we have before us is to find the complete set of initial conditions

$$\begin{aligned} x(0) &= c, \\ \alpha(0) &= \alpha_0. \end{aligned} \tag{5}$$

such that the solution of the nonlinear system

$$\begin{aligned} \dot{x} &= f(x,\alpha) \\ \dot{\alpha} &= 0, \end{aligned} \tag{6}$$

agrees with the boundary conditions

$$x(t_i) = b_i, \tag{7}$$

where b_i is the observed state of the system at time t_i. Let us suppose that we have exactly $R = N + M$ measurements of the first component of x, where N is the dimension of x and M is the dimension of α.

The boundary conditions are readily modified for a two point boundary value problem, or for more than R observations, or for other types of measurements, for example linear combinations of the components of x.

Our approach to the problem is one of successive approximations. We solve a sequence of linear problems. We assume only that large systems of ordinary differential equations, whether linear or nonlinear, may be accurately integrated numerically if initial conditions are prescribed, and that linear algebraic systems may be accurately resolved.

Let us define a new column vector x of dimension R, having as its elements the components of the original vector x and the components of α,

$$(8) \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \\ \\ \cdot \\ \cdot \\ \cdot \\ \\ \\ x_R \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_N \\ \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_M \end{bmatrix} .$$

This vector of dependent variables x(t) satisfies the system of nonlinear equations

$$(9) \qquad \dot{x} = f(x)$$

according to (6), and it has the unknown initial conditions

$$(10) \qquad x(0) = c,$$

according to (5). The boundary conditions are

$$(11) \qquad x_1(t_i) = b_i, \quad i = 1, 2, \ldots, R.$$

Mathematically, we need not distinguish between the components of this new vector x as state variables or system parameters.

An initial approximation starts the calculations. We form an estimate of the initial vector c, and we integrate system (9) to produce the solution $x(t)$ over the time interval of interest, $0 \leq t \leq T$, via numerical integration. The quasilinearization procedure is applied iteratively until a convergence to a solution occurs, or the solution diverges. Let us suppose that we have completed stage k of our calculations and we have the current approximation $x^k(t)$. In stage $k + 1$, we wish to calculate a new approximation $x^{k+1}(t)$.

The vector function $x^{k+1}(t)$ is the solution of the linear system

$$(12) \qquad \dot{x}^{k+1} = f(x^k) + J(x^k)\,(x^{k+1} - x^k),$$

where $J(x)$ is the Jacobian matrix with elements

$$(13) \qquad J_{ij} = \frac{\partial f_i}{\partial x_j}.$$

Since x^{k+1} is a solution of a system of linear differential equations, we know from general theory that it may be represented as the sum of a particular solution, $p(t)$, and a linear combination of R independent solutions of the homogeneous equations, $h^i(t)$, $i = 1, 2, \ldots, R$,

(14) $$x^{k+1}(t) = p(t) + \sum_{i=1}^{R} c^i h^i(t).$$

The function p satisfies the equation

(15) $$\dot{p} = f(x^k) + J(x^k)(p - x^k),$$

and for convenience we choose the initial conditions

(16) $$p(0) = 0.$$

The functions h^i are solutions of the homogeneous systems

(17) $$\dot{h}^i = J(x^k) h^i,$$

and we choose the initial conditions

(18) $h^i(0)$ = the unit vector with all of its components zero, except for the i^{th} which is one.

The $h^i(0)$ form a linearly independent set. If the interval $(0,T)$ is sufficiently small, the functions $h^i(t)$ are also independent. The solutions $p(t)$, $h^i(t)$ are produced by numerical integration with the given initial conditions. There are $R+1$ systems of differential equations, each with R equations, making a total of $R(R+1)$ equations which are integrated at each stage of our calculations.

After the functions p and h^i have been found over the interval, we must combine them so as to satisfy the boundary conditions (11),

(19) $$b_i = p_1(t_i) + \sum_{j=1}^{R} c^j h_1^j(t_i), \quad i = 1, 2, \ldots, R.$$

This results in a system of R linear algebraic equations for the determination of the R unknown multipliers c^j, of the standard form

(20) $$A c = B,$$

where the elements of the $R \times R$ matrix of coefficients A are

(21) $$A_{ij} = h_1^j(t_i),$$

and the components of the R-dimensional column vector B are

(22) $$B_i = b_i - p_1(t_i).$$

Having determined the multipliers, we now know a complete set of initial conditions for the $(k+1)^{st}$ stage.

(23) $$c = x^{k+1}(0) = p(0) + \sum_{j=1}^{R} c^j h^j(0).$$

Because of our choice of initial conditions for p and h^j, the initial values for each component of the vector x are identical with the multipliers c^j,

(24) $$c_i = x_i^{k+1}(0) = c^i, \quad i = 1, 2, \ldots, R.$$

Furthermore, we have a new approximation to the system parameter vector α.

(25) $\quad z_i = c^{N+i}, \quad i = 1, 2, \ldots, M.$

The new approximation $x^{k+1}(t)$ for the interval $(0,T)$ may be produced either by the integration of the linear equations with the initial conditions just found, or by the linear combination of $p(t)$ and $h(t)$. The $(k+1)^{st}$ cycle is complete and we are ready for the $(k+2)^{nd}$. The process may be repeated until no further change is noted in the vector c.

The quasilinearization procedure is analogous to Newton's method for finding roots of an equation, $f(x) = 0$. If x^0 is an approximate value of one of the roots of $f(x)$, then an improved value x^1 is obtained by applying the Taylor expansion formula to $f(x)$, and neglecting higher derivatives,

(26) $$f(x^1) = f(x^0) + (x^1 - x^0) \frac{\partial f(x^0)}{\partial x^0} .$$

Thus, the next approximation of the root is

(27) $$x^1 = x^0 - \frac{f(x^0)}{f'(x^0)} .$$

In quasilinearization, if the function $x^0(t)$ is an approximate solution of the nonlinear differential equation,

(28) $$\dot{x} = f(x) .$$

then an improved solution $x^1(t)$ may be obtained in the following manner. The function $f(x)$ is expanded around the current estimate $x^0(t)$, neglecting higher derivatives,

$$f(x^1) = f(x^0) + (x^1 - x^0)\frac{\partial f(x^0)}{\partial x^0}. \tag{29}$$

The improved approximation $x^1(t)$ is the solution of the linear equation,

$$\dot{x}^1 = f(x^0) + (x^1 - x^0)\frac{\partial f(x^0)}{\partial x^0}. \tag{30}$$

The method is easily extended to vector functions, as we have seen. The sequence of functions $x^1(t)$, $x^2(t)$, $x^3(t)$, ... may be shown to converge quadratically in the limit[17]. Practically speaking, a good initial approximation leads to rapid convergence, with the number of correct digits approximately doubling with each additional iteration. On the other hand, a poor initial approximation may lead to divergence.

The quasilinearization technique provides a systematic way of treating nonlinear boundary value problems. The computational solution of such a problem is broken up into stages, in which a large system of ordinary differential equations is integrated with known initial conditions, and a linear algebraic system is resolved. The initial value integration problem is well-suited to the digital computer. With the aid of a formula such as the trapezoidal rule,

$$\int_{t_0}^{t_n} f(t)dt \cong \frac{\Delta}{2}(f_0 + f_1) + \frac{\Delta}{2}(f_1 + f_2) + \ldots + \frac{\Delta}{2}(f_{n-1} + f_n). \tag{31}$$

the integral of a function over an interval is rapidly computed. Moreover, higher order methods such as the Runge-Kutta and the Adams-Moulton, usually of fourth order, make it possible to solve the integration problem accurately and rapidly. The accuracy may be as high as one part in 10^8. The solution is available at each grid point t_0, $t_0 + \Delta$, $t_0 + 2\Delta$, ... t_n, and may be stored in the computer's memory for use at some future time. The rapid-access storage capability of a computer such as the IBM 7090 or 7044 is 32,000 words. The integration of several hundred first order equations is a routine affair.

On the other hand, the solution of a linear algebraic system is not a routine matter, computationally speaking. While formulas exist for the numerical inversion of a matrix, the solution may be inaccurate. The matrix may be ill-conditioned, and other techniques may have to be brought into play to remedy the situation [20]. The storage of the n^{th} approximation for the calculation of the $(n + 1)^{st}$ approximation may become a problem; a suggestion for overcoming this difficulty is given in [21].

4. SOLUTION OF THE POTENTIAL PROBLEM

We follow the method of quasilinearization to identify the unknown mass and to solve the problem of potential determination of Section 2. The nonlinear system of equations is

$$\ddot{x} = -\frac{x}{r^3} - \alpha \frac{x-\xi}{s^3}$$

(1) $$\ddot{y} = -\frac{y}{r^3} - \alpha \frac{y-\eta}{s^3}$$

$$\dot{\alpha} = 0$$

with

(2) $$r^2 = x^2 + y^2 \; , \quad s^2 = (x-\xi)^2 + (y-\eta)^2 \; .$$

System (1) is equivalent to a system of five first order equations for x, $\dot{x}$, y, $\dot{y}$, and α. The system of linear equations for the $(k+1)^{st}$ stage is

(3)
$$\begin{aligned}
\ddot{x}^{k+1} &= \left\{-\frac{x^k}{r^3} - \alpha^k \frac{x^k-\xi}{s^3}\right\} \\
&+ (x^{k+1}-x^k)\left\{-\frac{1}{r^3} + \frac{3x^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(x^k-\xi)^2}{s^5}\right\} \\
&+ (y^{k+1}-y^k)\left\{\frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k-\xi)(y^k-\eta)}{s^5}\right\} \\
&+ (\alpha^{k+1}-\alpha^k)\left\{-\frac{x^k-\xi}{s^3}\right\} \; , \\
\ddot{y}^{k+1} &= \left\{-\frac{y^k}{r^3} - \alpha^k \frac{y^k-\eta}{s^3}\right\} \\
&+ (x^{k+1}-x^k)\left\{\frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k-\xi)(y^k-\eta)}{s^5}\right\} \\
&+ (y^{k+1}-y^k)\left\{-\frac{1}{r^3} + \frac{3y^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(y^k-\eta)^2}{s^5}\right\} \\
&+ (\alpha^{k+1}-\alpha^k)\left\{-\frac{y^k-\eta}{s^3}\right\} \; , \\
\dot{\alpha}^{k+1} &= 0 \; ,
\end{aligned}$$

where

(4) $$r^2 = (x^k)^2 + (y^k)^2 \; , \quad s^2 = (x^k-\xi)^2 + (y^k-\eta)^2 \; .$$

We express the solution of (3) as the sum of a particular solution of (3) plus a linear combination of five independent solutions of the homogeneous form of (3),

$$
\begin{aligned}
x^{k+1}(t) &= p_x(t) + \sum_{j=1}^{5} c^j\, h_x^j(t)\ , \\
y^{k+1}(t) &= p_y(t) + \sum_{j=1}^{5} c^j\, h_y^j(t)\ , \\
\alpha^{k+1}(t) &= p_\alpha(t) + \sum_{j=1}^{5} c^j\, h_\alpha^j(t)\ .
\end{aligned} \tag{5}
$$

Here, the symbol $p_x(t)$ is meant to represent the x component of the particular solution, which is a vector of dimension five, and similarly for the symbols $p_y(t)$, $p_\alpha(t)$. The symbols $h_x^j(t)$, $h_y^j(t)$, $h_\alpha^j(t)$ respectively correspond to the x, y, and α components of the j^{th} homogeneous solution vector, for $j = 1, 2, \ldots, 5$. The system which the particular solution satisfies is

$$
\begin{aligned}
\ddot{p}_x = &\left\{ -\frac{x^k}{r^3} - \alpha^k \frac{x^k-\xi}{s^3} \right\} \\
&+ (p_x - x^k)\left\{ -\frac{1}{r^3} + \frac{3x^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(x^k-\xi)^2}{s^5} \right\} \\
&+ (p_y - y^k)\left\{ \frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k-\xi)(y^k-\eta)}{s^5} \right\} \\
&+ (p_\alpha - \alpha^k)\left\{ -\frac{x^k-\xi}{s^3} \right\}\ .
\end{aligned} \tag{6}\ldots
$$

$$
\begin{aligned}
\ddot{p}_y = &\left\{ -\frac{y^k}{r^3} - \alpha^k \frac{y^k-\eta}{s^3} \right\} \\
&+ (p_x - x^k)\left\{ \frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k-\xi)(y^k-\eta)}{s^5} \right\}
\end{aligned}
$$

$$+ (p_y - y^k) \left\{ -\frac{1}{r^3} + \frac{3y^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(y^k - \eta)^2}{s^5} \right\}$$

$$(6) \qquad + (p_\alpha - \alpha^k) \left\{ -\frac{y^k - \eta}{s^3} \right\} ,$$

$$\dot{p}_\alpha = 0 .$$

with the initial condition

$$(7) \qquad p(0) = 0 .$$

The j^{th} homogeneous solution satisfies the system

$$\ddot{h}_x^j = h_x^j \left\{ -\frac{1}{r^3} + \frac{3x^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(x^k - \xi)^2}{s^5} \right\}$$

$$+ h_y^j \left\{ \frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k - \xi)(y^k - \eta)}{s^5} \right\}$$

$$+ h_\alpha^j \left\{ -\frac{x^k - \xi}{s^3} \right\}$$

$$(8) \qquad \ddot{h}_y^j = h_x^j \left\{ \frac{3x^k y^k}{r^5} + \frac{3\alpha^k(x^k - \xi)(y^k - \eta)}{s^5} \right\}$$

$$+ h_y^j \left\{ -\frac{1}{r^3} + \frac{3y^{k2}}{r^5} - \frac{\alpha^k}{s^3} + \frac{3\alpha^k(y^k - \eta)^2}{s^5} \right\}$$

$$+ h_\alpha^j \left\{ -\frac{y^k - \eta}{s^3} \right\} .$$

$$\dot{h}_\alpha^j = 0 .$$

Its five initial conditions are presented in the appropriate column of Table 1.

TABLE I.

THE INITIAL CONDITIONS FOR THE HOMOGENEOUS SOLUTIONS

	j=1	2	3	4	5
$h_x^j(0)$	1	0	0	0	0
$h_{\dot{x}}^j(0)$	0	1	0	0	0
$h_y^j(0)$	0	0	1	0	0
$h_{\dot{y}}^j(0)$	0	0	0	1	0
$h_\alpha^j(0)$	0	0	0	0	1

The particular and homogeneous solutions are produced by numerical integration and are known at the discrete times $t = 0, \Delta, 2\Delta, 3\Delta, \ldots, T$.

Let us find the system of linear algebraic equations which is to be solved in the $(k+1)^{st}$ stage. The boundary conditions may be expressed as

$$(9) \qquad y^{k+1}(t_i) + [1-x^{k+1}(t_i)] \tan\theta(t_i) = 0 ,$$

where $\theta(t_i)$ is the observed angular position of the heavenly body H at time t_i. Using relations (5), we obtain the five equations

$$(10)\ldots \qquad \begin{aligned} &\sum_{j=1}^{5} c^j \, [h_y^j(t_i) - h_x^j(t_i) \tan\theta(t_i)] \\ &= -\tan\theta(t_i) - p_y(t_i) + p_x(t_i) \tan\theta(t_i) , \end{aligned}$$

(10) $\quad i = 1, 2, \ldots, 5.$

for the five unknowns $c^1, c^2, \ldots c^5$.

The solution of (10) immediately gives us our new set of orbital parameters and the mass of the unknown perturbing body P,

$$
\begin{aligned}
x^{k+1}(0) &= c^1 , \\
\dot{x}^{k+1}(0) &= c^2 , \\
y^{k+1}(0) &= c^3 , \\
\dot{y}^{k+1}(0) &= c^4 , \\
\alpha^{k+1}(0) &= c^5 .
\end{aligned}
\tag{11}
$$

Since we need $x^{k+1}(t)$, $y^{k+1}(t)$, and $\alpha^{k+1}(t)$, for stage $k+2$, we use relations (5) for the evaluation of these functions at $t = 0, \Delta, 2\Delta, 3\Delta, \ldots, T$. The cycle is ready to begin once more, and it is repeated until a solution of the nonlinear problem is found, or for a fixed number of stages.

We begin a numerical experiment with the initial guess that at time $t = 0$, the body H is at location (3,0) with velocity coordinates $\dot{x} = 0$, $\dot{y} = 1$, and we believe that the mass of P is about 0.3. We integrate equations (1) with the initial values

$$x(0) = 3,\ \dot{x}(0) = 0,\ y(0) = 0,\ \dot{y}(0) = 1,\ \alpha(0) = 0.3 , \tag{12}$$

from $t = 0$ to $t = 2.5$, using a grid size of $\Delta = 0.01$ and an Adams-Moulton integration formula. This generates the curve labelled "Initial Approximation" in Fig. 2. This is a very poor approximation to the true orbit. After two stages of the quasilinearization scheme, our approximation has improved so that the orbit is represented by the curve labelled "Approximation 2" in Fig. 2. In five iterations, we converge to the true curve, $h(x,y)$, and we have found the correct value of 0.2 for the mass of the perturbing body. The rate of convergence is indicated in Table 2.

TABLE 2.

SUCCESSIVE APPROXIMATIONS OF THE COMPLETE SET OF INITIAL CONDITIONS AND THE MASS OF P

Approx.	$x(0)$	$\dot{x}(0)$	$y(0)$	$\dot{y}(0)$	α
0	3.0	0.0	0.0	1.0	0.3
1	3.18421	.221272	0.0	1.06544	-.120164
2	2.37728	-.061370	0.0	0.690767	-.259144
3	2.11189	-.018545	0.0	0.555666	-.070333
4	2.01974	-.003194	0.0	0.509813	.141922
5	2.00023	.000013	0.0	0.500120	.198208
True	2.0	0.0	0.0	0.5	0.2

Suppose that we also wish to know the position

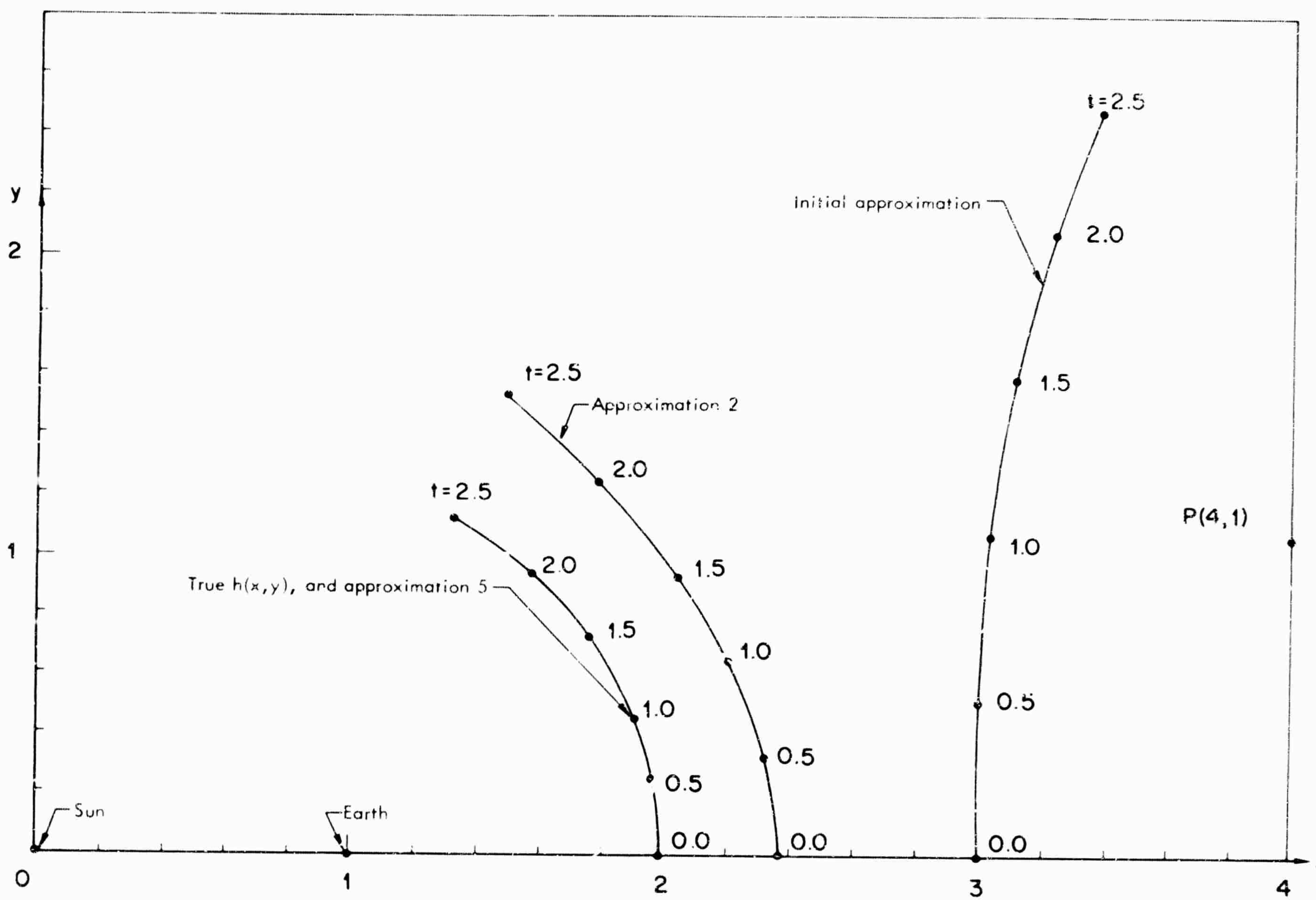


Fig. 2. Successive approximations of the orbit.

TABLE 3.

PREDICTED LOCATION OF H AT TIME 2.5

Approx.	x(2.5)	y(2.5)
0	3.38098	2.47759
1	1.93764	2.72562
2	1.48932	1.53066
3	1.37202	1.21823
4	1.34124	1.1251[illegible]
5	1.33503	1.10598
True	1.33494	1.10571

of H at some "future" time $t = 2.5$. Our sequence of approximations of the predicted location is given in Table 3. The entire calculations require only 1–1/2 minutes on the IBM 7044 computer, using a FORTRAN IV source language. The FORTRAN programs which generate the data and which determine the orbit and mass are listed in Appendix A.

The time involved is mainly due to the evaluation of the derivatives of the functions. The Adams–Moulton fourth order method requires the derivative to be evaluated twice for each integration step forward [25].

In another trial, beginning with the approximation that the orbit is a point at the earth's center [19], we find another solution which satisfies all of the conditions. However, the mass turns out to be greater than one, an unallowed solution. Repeating the experiment with more closely spaced observations, we converge to the true solution. The determination of the optimal set of observations is itself an interesting question.

REFERENCES

1. Bellman, R., H. Kagiwada and R. Kalaba, Quasilinearization, System Identification, and Prediction, The RAND Corporation, RM-3812-PR, August 1963. (To appear in Int. J. Eng. Sci.)

2. Bellman, R., C. Collier, H. Kagiwada, R. Kalaba and R. Selvester, "Estimation of Heart Parameters using Skin Potential Measurements," Communications of the ACM, Vol. 7. pp. 666-668, 1964.

3. Bellman, R., H. Kagiwada, and R. E. Kalaba, On the Identification of Systems and The Unscrambling of Data - I: Hidden Periodicities, The RAND Corporation, RM-4285-PR, September 1964.

4. Agranovich, Z. S., and V. A. Marchenko, The Inverse Problem of Scattering Theory, Gordon and Breach, New York, 1963.

5. Cannon, J. R., "Determination of Certain Parameters in Heat Conduction Problems." J. Math. Anal. Appl., Vol. 8, No. 2, pp. 188-201, 1964.

6. Harris, J. L., "Diffraction and Resolving Power," J. Opt. Soc. Am., Vol. 54, No. 7, pp. 931-936, 1964.

7. King, Jean I. F., "Inversion by Slabs of Various Thickness," J. Atm. Sci., Vol. 21, pp. 324-326, 1964.

8. Maslennikov, M. V., "Uniqueness of the Solution of the Inverse Problem of the Asymptotic Theory of Radiation Transfer," J. Numerical Analysis and Mathematical Physics, Vol. 2, pp. 1044-1053, 1962.

9. Preisendorfer, R. W., "Application of Radiative Transfer Theory to Light Measurements in the Sea," L'Institute Geographique National, Paris, International Union of Geodesy and Geophysics, Monographie No. 10, June 1961.

10. Sims, A. R., "Certain Aspects of the Inverse Scattering Problem," J. Soc. Indus. Appl. Math., Vol. 5, pp. 183-205, 1957

11. Gel'fand, I. M. and B. M. Levitan, "On the Determination of a Differential Equation from its Spectral Function," Translations of the American Mathematical Society, Vol. 2, pp. 253-304, 1955.

12. Yamamoto, Giichi, and Atsushi Shimanuki, "The Determination of Lateral Diffusivity in Diabatic Conditions near the Ground from Diffusion Experiments," *J. Atm. Sci.*, Vol. 21, pp. 187-196, 1964.

13. Bellman, R., R. Kalaba, and G. M. Wing, "Invariant Imbedding and Mathematical Physics - I: Particle Processes," *J. Math. Phys.*, Vol. 1, pp. 280-308, 1960.

14. Bellman, R., R. Kalaba, and M. C. Prestrud, *Invariant Imbedding and Radiative Transfer in Slabs of Finite Thickness*, American Elsevier Publishing Co., New York, 1963.

15. Bellman, R., H. Kagiwada, R. Kalaba and S. Ueno, *Some Computational Aspects of Radiative Transfer in Finite Slabs*, American Elsevier Publishing Company, Inc., New York, to appear.

16. Bellman, R., and R. Kalaba, *Quasilinearization and Boundary Value Problems*, American Elsevier Publishing Co., New York, 1965.

17. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," *J. Math. and Mech.*, Vol. 8, pp. 519-574, 1959.

18. Bellman, R. *Dynamic Programming*, Princeton University Press, Princeton, 1957.

19. Bellman, R., H. Kagiwada and Robert Kalaba, "Orbit Determination as a Multi-Point Boundary-Value Problem and Quasilinearization", *Proc. Nat. Acad. Sci.* Vol. 48, pp. 1327-1329, 1962.

20. Bellman, R., R. Kalaba and J. Lockett, *Dynamic Programming and Ill-Conditioned Linear Systems - II*, RM-4201-PR, The RAND Corporation, July 1964.

21. Bellman, R., "Successive Approximations and Computer Storage Problems in Ordinary Differential Equations," *Comm. of the ACM*, Vol. 4, 222-223, 1961.

22. Bellman, R., H. H. Kagiwada, R. Kalaba, and M. C. Prestrud, *Invariant Imbedding and Time-Dependent Transport Processes*, American Elsevier Publishing Co., Inc., New York, 1964.

23. Bellman, R., H. Kagiwada, and R. Kalaba, "Wengert's Method for Partial Derivatives, Orbit Determination and Quasilinearization", Communications of the ACM, Vol. 8, pp. 231-232, 1965.

24. Bellman, R., H. Kagiwada, R. Kalaba, and R. Sridhar, Invariant Imbedding and Nonlinear Filtering Theory, The RAND Corporation, RM-4374-PR, December 1964.

25. Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, John Wiley and Sons, New York, 1962.

26. Milne, W. E., Numerical Solution of Differential Equations, John Wiley and Sons, New York, 1953.

27. Lyttleton, R. A., "A Short Method for the Discovery of Neptune," Monthly Notices of the Royal Astronomical Society of London, Vol. 118, pp. 551-559, 1958.

28. Dubyago, A. D., The Determination of Orbits, The Macmillan Company, New York, 1961 (Translation by R. D. Burke, G. Gordon, L. N. Rowell, and F. T. Smith).

29. Smart, W. M., Text-Book on Spherical Astronomy, Cambridge University Press, 1960.

30. Chandrasekhar, S., Radiative Transfer, Oxford University Press, London, 1950.

31. Sobolev, V. V., A Treatise on Radiative Transfer, D. Van Nostrand Company, Princeton, 1963.

32. Kourganoff V., Basic Methods in Transfer Problems, Oxford University Press, London, 1952.

33. Busbridge, I. W., The Mathematics of Radiative Transfer, Cambridge University Press, 1960.

34. Wing, G. M., An Introduction to Transport Theory, John Wiley and Sons, New York, 1962.

35. Aller, L. H., Astrophysics, The Atmospheres of the Sun and Stars, Ronald Press Company, 1963.

36. Greenstein, J. L. (ed.), Stellar Atmospheres, University of Chicago Press, Chicago, 1960.

37. Van de Hulst, H. C., "Scattering in a Planetary Atmosphere," Astrophys. Journal, Vol. 107, pp. 220-246, 1948.

38. Henyey, L. G. and J. L. Greenstein, "Diffuse Radiation in the Galaxy," Astrophys. Journal, Vol. 93, pp. 70-83, 1941.

CHAPTER TWO

INVERSE PROBLEMS IN RADIATIVE TRANSFER: LAYERED MEDIA

1. INTRODUCTION

Some inverse problems in radiative transfer are concerned with the estimation of the optical properties of an atmosphere based on measurements of diffusely reflected radiation. The location and the intensity of the source of radiation are known. We consider a plane-parallel medium which is composed of two layers. Our aim is to determine the optical thickness and the albedo of each layer, from knowledge of the input radiation and the diffusely reflected light.

First we discuss the concept of invariant imbedding, and we apply this technique to the derivation of the equation for the diffuse reflection function of an inhomogeneous slab with isotropic scattering. The inverse problem is stated in terms of the reflection function, and we formulate the problem as a nonlinear boundary value problem. We then show how the formalism of quasilinearization can be used to solve this problem. We conduct several numerical experiments for the determination of optical thicknesses and albedos of the layers. Computational results are presented, and the FORTRAN computer programs which produced the results are given in Appendix B.

2. INVARIANT IMBEDDING

The traditional approach to wave and particle transport processes leads to linear functional equations with boundary conditions. While linearity enables eigenfunction expansions to be made, one finds great difficulty in solving the equation of transfer. The integration of ordinary differential equations with given initial conditions is done extremely efficiently by digital computers. This suggests that problems be formulated in just this way, with the physical situation as the guide. Invariant imbedding provides a flexible manner in which to relate properties of one process to those of neighboring processes. This also leads to the generalized semigroup concept [1].

In a particle process, one is led by invariant imbedding to differential-integral equations for reflection and transmission functions. By the use of quadrature formulas [2], one reduces the equations from integral-differential form to approximate systems of ordinary differential equations. The wave equation, on the other hand, may be reduced to a system of ordinary differential equations in at least two ways:

(1) assume the time factor of the form $e^{i\omega t}$ and the problem simplifies to the steady state situation, or (2) use Laplace transform methods. Both alternatives are discussed in later sections.

Invariant imbedding is a useful formalism, theoretically and computationally speaking. Principles of invariance were first introduced by Ambarzumian in 1943 [3] and developed by Chandrasekhar [4]. The invariance concept was further extended and generalized by Bellman and Kalaba [5–10]. The form in which invariant imbedding is applied in these chapters is indicated by this example. Suppose that a neutron multiplication process takes place in a rod of length x [11]. The investigator wishes to know the reflected flux r for an input of one particle per second. Rather than study the processes within the rod extending from 0 to x, which would be quite difficult, the experimenter would like to vary the length of the rod and see how the reflected flux changes. The rod length is made a variable of the problem, so that $r = r(x)$. The original situation is imbedded in a class of similar cases, for all lengths of rod, and one obtains directly the reflected flux for a rod of any length including the length under investigation. This flux is rather easily computed and it is physically meaningful [15,16].

3. THE DIFFUSE REFLECTION FUNCTION FOR AN INHOMOGENEOUS SLAB

Consider an inhomogeneous, plane-parallel, non-emitting and isotropically scattering atmosphere of finite optical thickness τ_1. The optical properties depend only on τ, the optical distance from the lower boundary ($0 \leq \tau \leq \tau_1$). The physical situation is sketched in Fig. 1. Parallel rays of light of net flux π per unit area normal to their direction of propagation are incident on the upper surface, $\tau = \tau_1$. The direction is characterized by the parameter μ_0 ($0 < \mu_0 \leq 1$), which is the cosine of the angle measured from the downward normal to the surface. The bottom surface is a completely absorbing boundary, so that no light is reflected from it. This assumption is not essential to our discussion.

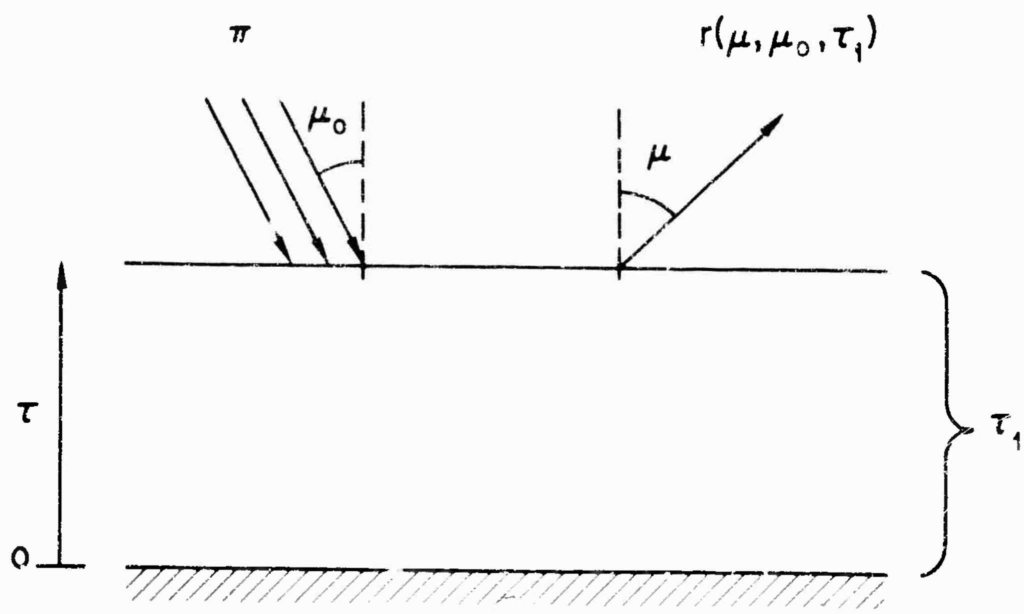


Fig. 1. Incident and reflected rays for an inhomogeneous slab of optical thickness τ_1.

The direction of the outgoing radiation is characterized by μ, the cosine of the polar angle measured from the outward normal to the top surface. This parameter is the direction cosine of the vector representing the ray of light. The azimuth angle has no significance due to the symmetry of the situation.

By "intensity" we shall mean the amount of energy which is transmitted through an element of area $d\sigma$ normal to the direction of flow, in a truncated elementary cone $d\omega$ in time dt. See Fig. 2, as well as Kourganoff [12]. We restrict ourselves to the steady-state situation at a fixed frequency.

We define the diffuse reflection function as follows:

(1) $r(\mu,\mu_0,\tau_1)$ is the intensity of the diffusely reflected light in the direction whose cosine is μ with respect to the outward normal, due to incident uniform parallel rays of radiation of constant net flux π in the direction whose cosine is μ_0 with respect to the inward normal, the slab having optical thickness τ_1.

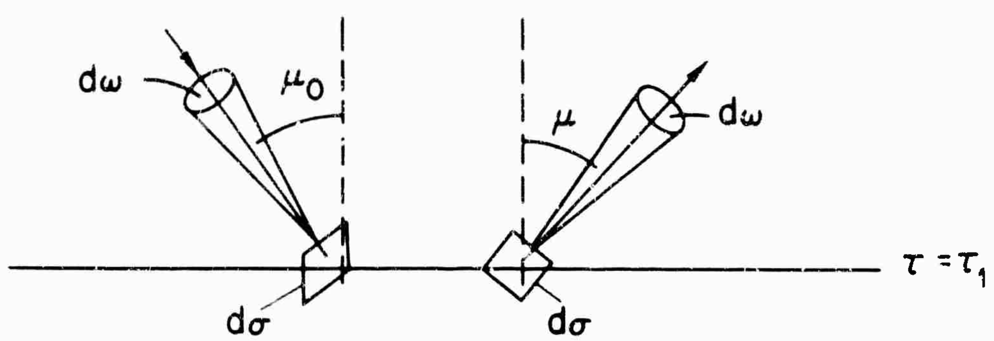


Fig. 2. The incident and reflected intensities.

We define a related function ρ,

$$(2) \quad \rho(\mu,\mu_0,\tau_1) = \frac{\mu \; r(\mu,\mu_0,\tau_1)}{\mu_0 \; \pi} ,$$

which is the energy of the diffusely reflected light in the direction μ passing through a unit of horizontal area per unit solid angle per unit time, due to incident radiation of unit energy per unit horizontal area per unit solid angle per unit time, in the direction μ_0. We may also interpret ρ as the probability that a particle will emerge from a unit of horizontal area at $\tau = \tau_1$, the top of a slab of thickness τ_1, going in direction μ, per unit solid angle per unit time, due to an input of one particle per unit horizontal area per unit solid angle per unit time in the direction μ_0.

Consider now a slab of thickness $\tau_1 + \Delta$ formed by adding a thin slab of thickness Δ to the top of the slab of thickness τ_1, as illustrated in Fig. 3. By imbedding the original problem in a class of problems of similar nature, we will derive an integro-differential equation for the diffuse reflection function.

The diffuse reflection function for a slab of thickness $\tau_1 + \Delta$ with an input of net flux π is $r(\mu,\mu_0, \tau_1 + \Delta) = \pi\rho(\mu,\mu_0,\tau_1 + \Delta)\mu_0/\mu$. Applying the method of invariant imbedding in its particle counting form to the probability of emergence of a particle from a slab, we obtain the equation

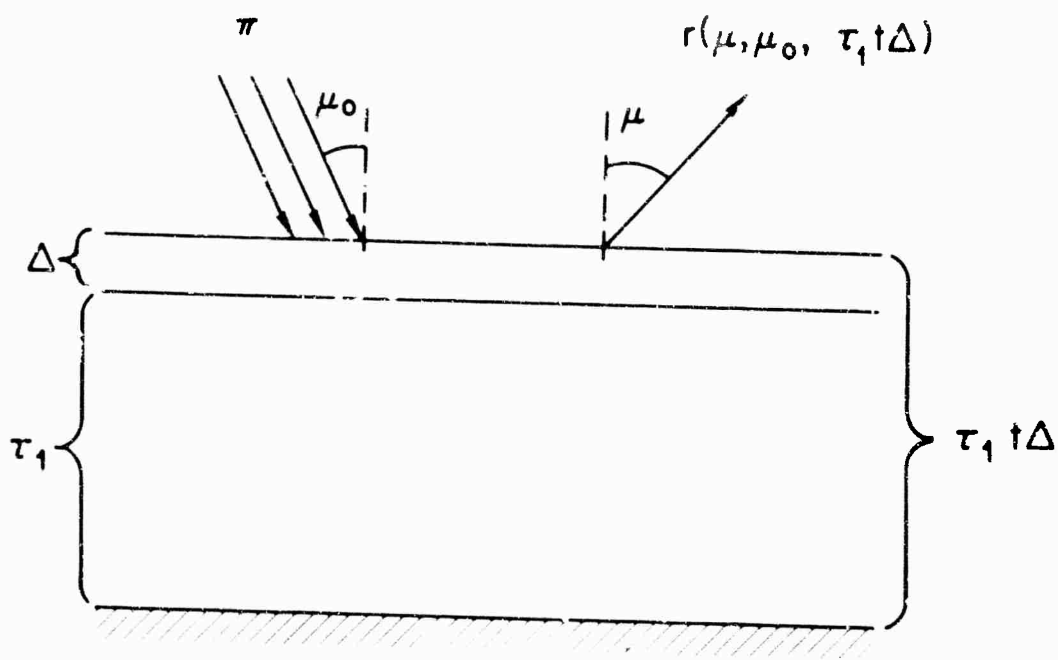


Fig. 3. An inhomogeneous slab of optical thickness $\tau_1 + \Delta$.

$$(3)\quad \rho(\mu,\mu_0,\tau_1+\Delta) = \rho(\mu,\mu_0,\tau_1) - \Delta\left(\frac{1}{\mu_0}+\frac{1}{\mu}\right)\rho(\mu,\mu_0,\tau_1)$$

$$+\frac{\Delta}{\mu_0}\frac{\lambda(\tau_1)}{4\pi} + 2\pi\int_0^1 \rho(\mu',\mu_0,\tau_1)d\mu'\,\frac{\Delta}{\mu'}\,\frac{\lambda(\tau_1)}{4\pi}$$

$$+ 2\pi\int_0^1 \frac{\Delta}{\mu_0}\,\frac{\lambda(\tau_1)}{4\pi}\; d\mu''\,\rho(\mu,\mu'',\tau_1)$$

$$+ 2\pi\int_0^1 \rho(\mu',\mu_0,\tau_1)d\mu'\,\frac{\Delta}{\mu'}\,\frac{\lambda(\tau_1)}{4\pi}\,2\pi\int_0^1 \rho(\mu,\mu'',\tau_1)d\mu''.$$

$$+\, o(\Delta).$$

The first term on the right handside is the probability that a particle emerges without any interaction in the thin slab. The unit of distance is such that x is the probability of an interaction in a path of length x. Hence the second term represents the losses due to interactions of the incoming and outgoing particles whose path lengths in Δ are Δ/μ_0 and Δ/μ respectively. The third term is the probabili y of an interaction and re-emission isotropically into the direction given by μ. The function $\lambda(\tau_1)$ is the probability of re-emission, and is called the albedo for single scattering. The next term is the probability that the particle is diffusely reflected from the slab between $(0,\tau_1)$ into the direction μ' and interacts in Δ and is re-emitted into the direction of emergence μ. The next term is the probability that an incoming particle interacts in Δ, enters the slab $(0,\tau_1)$ and is diffusely scattered into the emergent direction μ. The sixth term is the probability of diffuse reflection in $(0,\tau_1)$, then interaction and re-emission in Δ, and diffuse reflection from $(0,\tau_1)$ with outgoing direction μ. All other probabilities are proportional to Δ^2 or higher powers of Δ and are accounted for in the term $o(\Delta)$.

Let the diffusely reflected intensity be given by a new function R, by means of the relation

(4) $$r(\mu,\mu_0,\tau_1) = \frac{R(\mu,\mu_0,\tau_1)}{4\mu} ,$$

where R is related to ρ by the formula

(5) $$\rho(\mu,\mu_0,\tau_1) = \frac{R(\mu,\mu_0,\tau_1)}{4\pi\mu_0} .$$

Then R satisfies the equation

(6) $$R(\mu,\mu_0,\tau_1 + \Delta) = R(\mu,\mu_0,\tau_1) - \Delta\left(\frac{1}{\mu_0} + \frac{1}{\mu}\right)R(\mu,\mu_0,\tau_1)$$
$$+ \Delta\lambda\left\{1 + \frac{1}{2}\int_0^1 R(\mu',\mu_0,\tau_1)\frac{d\mu'}{\mu'} + \frac{1}{2}\int_0^1 R(\mu,\mu'',\tau_1)\frac{d\mu''}{\mu''}\right.$$
$$\left.+ \frac{1}{4}\int_0^1 R(\mu',\mu_0,\tau_1)\frac{d\mu'}{\mu'}\int_0^1 R(\mu,\mu'',\tau_1)\frac{d\mu''}{\mu''}\right\} + o(\Delta).$$

We expand the lefthand side of the equation in powers of Δ,

(7) $$R(\mu,\mu_0,\tau_1 + \Delta) = R(\mu,\mu_0,\tau_1) + \frac{\partial R(\mu,\mu_0,\tau_1)}{\partial\tau_1}\Delta + o(\Delta).$$

By letting $\Delta \to 0$, we arrive at the integro-differential equation

(8)
$$\frac{\partial R(\mu,\mu_0,\tau_1)}{\partial \tau_1} + \left(\frac{1}{\mu_0} + \frac{1}{\mu}\right) R$$
$$= \lambda(\tau_1)\left[1 + \frac{1}{2}\int_0^1 R(\mu',\mu_0,\tau_1)\,\frac{d\mu'}{\mu}\right]$$
$$\cdot \left[1 + \frac{1}{2}\int_0^1 R(\mu,\mu'',\tau_1)\,\frac{d\mu''}{\mu''}\right] .$$

The initial condition is

(9) $$R(\mu,\mu_0,0) = 0,$$

because no light is diffusely reflected when the medium has zero thickness. It is readily seen that the function R is symmetric [4,13,14,17], i.e.,

(10) $$R(\mu,\mu_0,\tau_1) = R(\mu_0,\mu,\tau_1).$$

Equation (8) for R is the same as Chandrasekhar's equation for his scattering function S, when the medium is homogeneous and isotropic.

4. GAUSSIAN QUADRATURE

The above integrals may be evaluated by the use of Gaussian quadrature [4,13,14]. Since the limits of our integrals are zero to one, we use the approximate relation

$$(1) \qquad \int_0^1 f(x)\,dx \cong \sum_{k=1}^{N} f(a_k) w_k ,$$

which is exact if $f(x)$ is a polynomial of degree $2N-1$ or less. The numbers a_k are roots of the shifted Legendre function $P_N^*(x) = P_N(1-2x)$ on the interval $(0,1)$, and the numbers w_k are the corresponding weights. For a more detailed discussion and for tables of roots and weights, see [13].

Replacing integrals by Gaussian sums, we have the following equation which is approximately true,

$$(2) \qquad \frac{\partial R(\mu,\mu_0,\tau_1)}{\partial \tau_1} + \left(\frac{1}{\mu_0} + \frac{1}{\mu}\right) R =$$

$$= \lambda(\tau_1) \left[1 + \frac{1}{2} \sum_{k=1}^{N} R(\mu_k,\mu_0,\tau_1) \frac{w_k}{\mu_k}\right]\left[1 + \frac{1}{2} \sum_{k=1}^{N} R(\mu,\mu_k,\tau_1) \frac{w_k}{\mu_k}\right] .$$

For $N \sim 7$, this is a fairly good approximation [14,15]. We consider only those incident and outgoing directions for which the cosines take on the values of the roots μ_k.

For $N = 7$, the roots μ_k and the corresponding angles, arc cosine μ_k, are listed in Table 1, in order of increasing μ.

TABLE 1

ROOTS OF SHIFTED LEGENDRE POLYNOMIALS OF DEGREE N = 7, AND CORRESPONDING ANGLES

k	Roots μ_k	Arc cosine μ_k (in degrees)
1	0.025446044	88.541891
2	0.12923441	82.574646
3	0.29707742	72.717849
4	0.50000000	60.000000
5	0.70292258	45.338044
6	0.87076559	29.452271
7	0.97455396	12.953079

We define the functions of one argument,

(3) $$R_{ij}(\tau_1) = R(\mu_i, \mu_j, \tau_1),$$

for $i = 1,2,\ldots,N$, $j = 1,2,\ldots,N$. Then (2) becomes a system of ordinary differential equations

(4) $$\frac{dR_{ij}(\tau_1)}{d\tau_1} + \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)R_{ij} =$$

$$\lambda(\tau_1)\left[1 + \frac{1}{2}\sum_{k=1}^{N} R_{kj}(\tau_1)\frac{w_k}{\mu_k}\right]\left[1 + \frac{1}{2}\sum_{k=1}^{N} R_{ik}(\tau_1)\frac{w_k}{\mu_k}\right],$$

with optical thickness τ_1 as the independent variable. The initial conditions are, of course,

(5) $$R_{ij}(0) = 0.$$

The system of N^2 first order differential equations reduce to a system of $N(N+1)/2$ equations by the use of the symmetry property of R. This is a large saving of computational effort.

5. AN INVERSE PROBLEM

Consider the inhomogeneous medium composed of two layers as illustrated in Fig. 4.

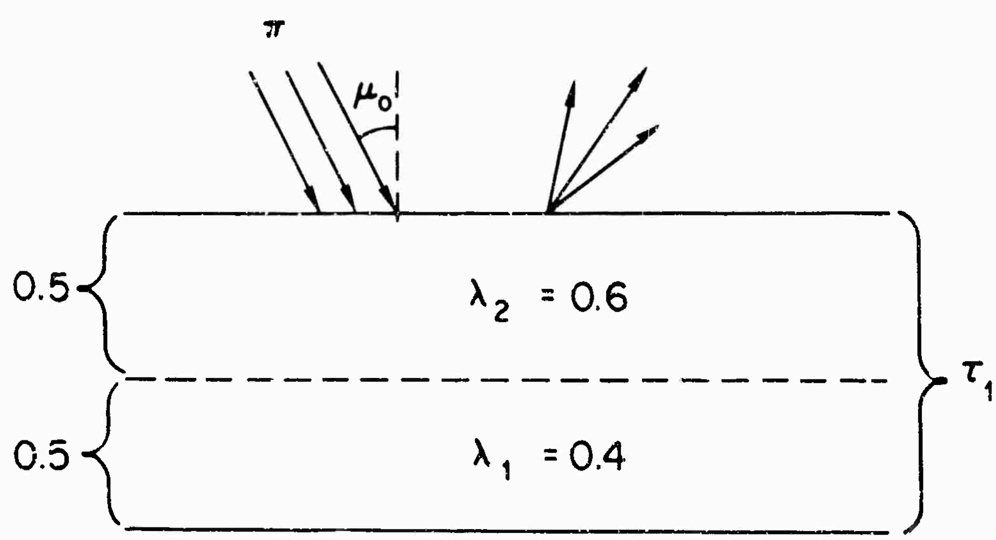


Fig. 4. A layered medium

The total thickness of the medium is 1.0, the thickness of each slab is 0.5, and the albedos are 0.4 in the lower layer, 0.6 in the upper layer. In order to have a continuous function for the albedo, we assume that λ is given by the function

(1) $$\lambda(\tau) = 0.5 + 0.1 \tanh 10(\tau-0.5).$$

This function is plotted in Fig. 5.

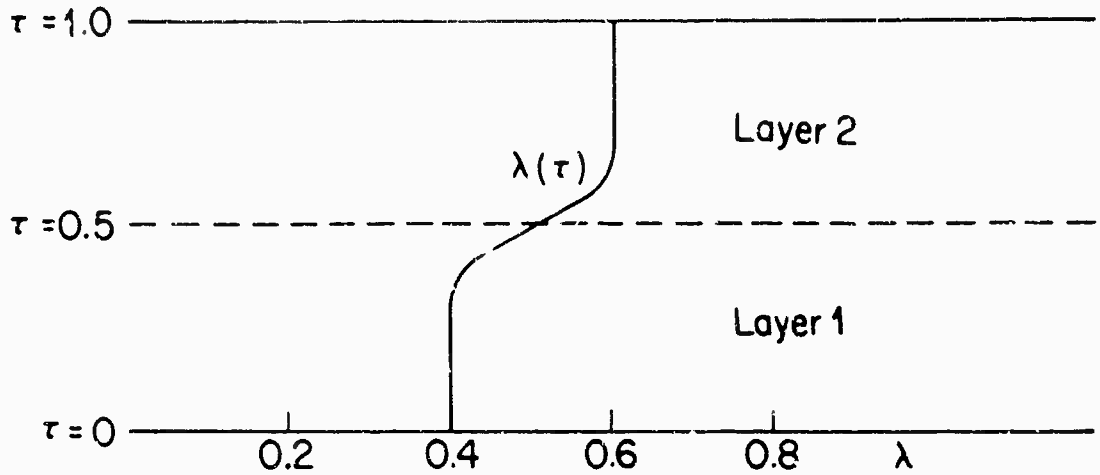


Fig. 5. The albedo function $\lambda(\tau) = 0.5 + 0.1 \tanh 10(\tau - 0.5)$ for a slab of thickness 1.0.

Parallel rays of net flux π are incident on the upper surface of the medium in a direction characterized by the parameter μ_j. We obtain N^2 measurements of the intensity of the diffusely reflected light, $b_{ij} \cong r_{ij}(\tau_1)$, for incident directions μ_j, $j = 1,2,\ldots,N$, and reflection directions μ_i, $i = 1,2,\ldots,N$. We wish to determine the nature of the medium from the knowledge of the reflected radiation.

Let the total optical thickness of the slab be T, and let the thickness of the lower layer be c. Let the albedos be λ_1 and λ_2, for the lower and upper slabs respectively, where the albedo as a function of optical elevation is

(2) $\lambda(\tau) = a + b \tanh 10(\tau - c)$

and $\lambda_1 \cong a-b,$

(3) $\lambda_2 \cong a+b,$

where a and b are unknown parameters. For the "true" situation,

(4) $T = 1.0,\ a = 0.5,\ b = 0.1,\ c = 0.5.$

The inverse problem which we wish to solve is to determine the quantities T, a, b, and c in such a way as to have best agreement, in the least square sense, between the estimated solution using the ordinary differential equations for r_{ij} and the observed reflection pattern. Mathematically speaking, we wish to minimize the expression

(5) $$\sum_{i=1}^{N} \sum_{j=1}^{N} \{r_{ij}(T) - b_{ij}\}^2$$

over all choices of the unknown parameters.

In Table 2, we present the measurements $\{b_{ij}\}$ for $N = 7$. In Fig. 6 we plot some of the measurements as a function of the cosine of the reflection angle, $\mu \sim \mu_i$, for input directions $\mu_0 \sim \mu_j \cong .025$, .5, and .975. The discrete observations are shown as dots, and for clarity we draw smooth curves through these points. For comparison, we show what the corresponding measurements would be if the medium were homogeneous with albedo $\lambda = 0.5$.

TABLE 2

THE MEASUREMENTS $\{b_{ij}\}$

	i = 1	2	3	4	5	6	7
j = 1	0.079914	0.028164	0.014304	0.009104	0.006707	0.005515	0.004970
2	0.143038	0.091522	0.058437	0.040826	0.031405	0.026378	0.023989
3	0.167000	0.134331	0.099653	0.075106	0.060044	0.051445	0.047248
4	0.178898	0.157955	0.126408	0.099392	0.081253	0.070435	0.065042
5	0.185284	0.170817	0.142072	0.114229	0.094495	0.082423	0.076332
6	0.188723	0.177733	0.150791	0.122665	0.102104	0.089349	0.082870
7	0.190354	0.180898	0.154995	0.126773	0.105829	0.092748	0.086083

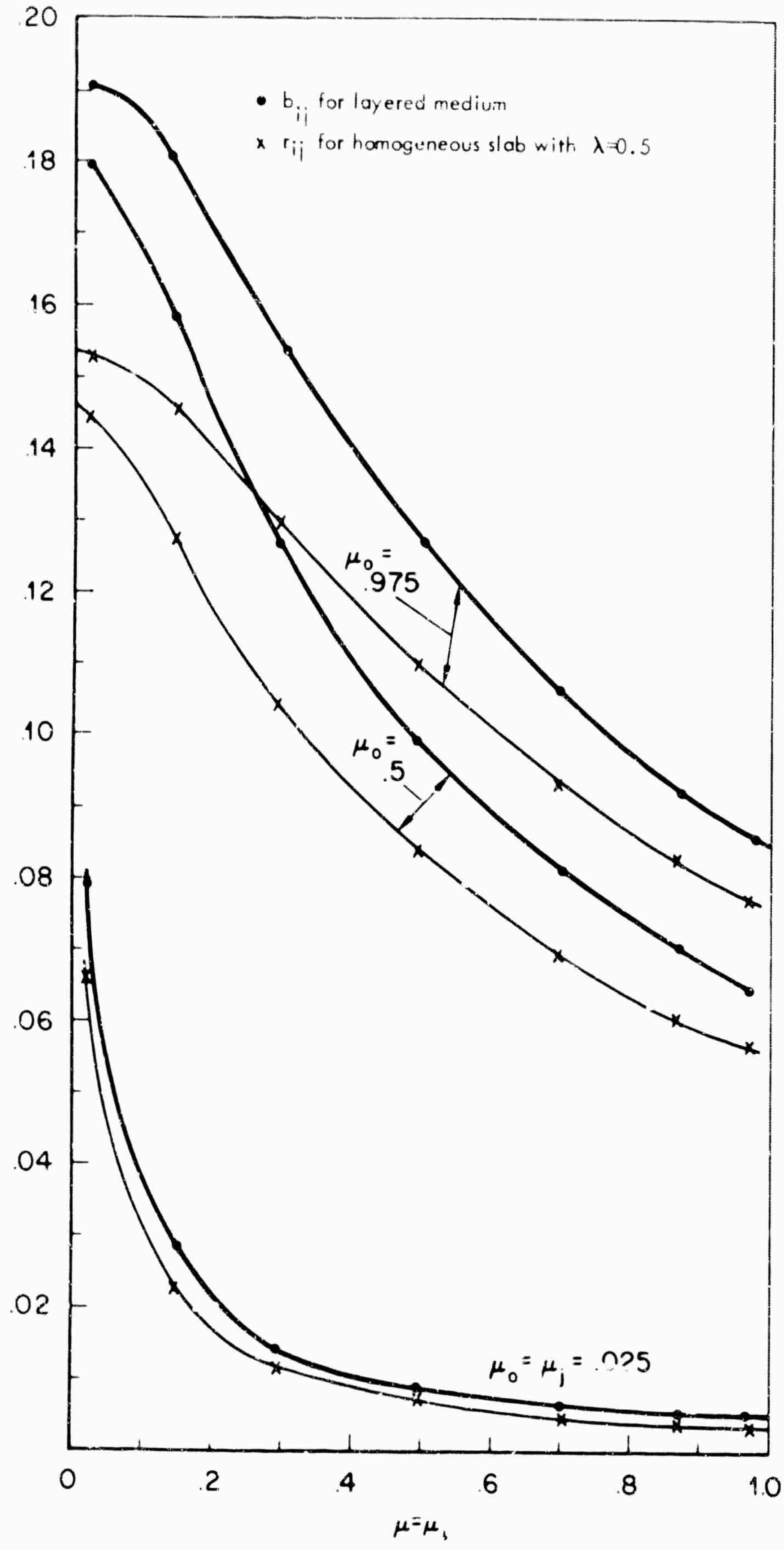


Fig. 6. Some of the measurements $\{b_{ij}\}$ for a layered medium.

6. FORMULATION AS A NONLINEAR BOUNDARY VALUE PROBLEM

We formulate this inverse problem as a nonlinear boundary value problem. To the system of N^2 nonlinear differential equations

$$(1)\quad \frac{dR_{ij}}{d\tau_1} + \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)R_{ij} =$$

$$\lambda(\tau_1)\ \left[1 + \frac{1}{2}\sum_{k=1}^{N} R_{kj}\frac{w_k}{\mu_k}\right]\left[1 + \frac{1}{2}\sum_{k=1}^{N} R_{ik}\frac{w_k}{\mu_k}\right],$$

where

$$(2)\qquad \lambda(\tau_1) = a + b \tanh 10(\tau_1 - c),$$

we add the differential equations

$$(3)\quad \frac{da}{d\tau_1} = 0,\quad \frac{db}{d\tau_1} = 0,\quad \frac{dc}{d\tau_1} = 0,\quad \frac{dT}{d\tau_1} = 0$$

because a, b, and c and T are unknown constants. The boundary conditions are

$$(4)\qquad R_{ij}(0) = 0,$$

and

$$(5)\quad \frac{\partial S}{\partial a} = 0,\quad \frac{\partial S}{\partial b} = 0,\quad \frac{\partial S}{\partial c} = 0,\quad \frac{\partial S}{\partial T} = 0,$$

where

$$(6)\quad S = \sum_{i=1}^{N}\ \sum_{j=1}^{N} \{R_{ij}(T) - 4\mu_i b_{ij}\}^2 .$$

7. NUMERICAL EXPERIMENTS I. DETERMINATION OF c, THE THICKNESS OF THE LOWER LAYER

Let us try to determine the quantity c, which is the thickness of the lower layer of the stratified medium. We assume that all of the other parameters a,b, and T are known. The parameter c is considered to be a function of optical height τ_1 described by the equation $dc/d\tau_1 = 0$. By following the method of quasilinearization as described previously, we obtain a system of linear differential equations for the $(k+1)^{st}$ approximation to R_{ij} and c:

$$(1) \quad \frac{dR_{ij}^{k+1}}{d\tau_1} = f(R_{ij}^k, c^k) + \sum_{i,j} (R_{ij}^{k+1} - R_{ij}^k) \frac{\partial f}{\partial R_{ij}^k}$$

$$+ (c^{k+1} - c^k) \frac{\partial f}{\partial c^k},$$

$$\frac{dc^{k+1}}{d\tau_1} = 0,$$

where

$$(2) \quad f(R_{ij}^k, c^k) = -\left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R_{ij}^k + \lambda(c^k)\left(1 + \frac{1}{2}\sum_{\ell=1}^{N} R_{\ell j}^k \frac{w_\ell}{\mu_\ell}\right)$$

$$\cdot \left(1 + \frac{1}{2}\sum_{\ell=1}^{N} R_{i\ell}^k \frac{w_\ell}{\mu_\ell}\right),$$

$$(3) \quad \lambda(c^k) = a + b \tanh 10(\tau_1 - c^k).$$

After simplifying, we have

$$(4)\quad \frac{dR_{ij}^{k+1}}{d\tau_1} = \Big\{-\left(\frac{1}{\mu_i}+\frac{1}{\mu_j}\right)R_{ij}^{k} + \lambda(c^k)\left(1+\frac{1}{2}\sum_{\ell=1}^{N} R_{i\ell}^{k}\frac{w_\ell}{\mu_\ell}\right) \times$$

$$\left(1+\frac{1}{2}\sum_{\ell=1}^{N} R_{\ell j}^{k}\frac{w_\ell}{\mu_\ell}\right)\Big\} + \Big\{-\left(\frac{1}{\mu_i}+\frac{1}{\mu_j}\right)(R_{ij}^{k+1}-R_{ij}^{k}) +$$

$$\frac{1}{2}\lambda(c^k)\Big[\left(1+\frac{1}{2}\sum_{\ell=1}^{N} R_{i\ell}^{k}\frac{w_\ell}{\mu_\ell}\right)\times\sum_{\ell=1}^{N}(R_{\ell j}^{k+1}-R_{\ell j}^{k})\frac{w_\ell}{\mu_\ell} +$$

$$\left(1+\frac{1}{2}\sum_{\ell=1}^{N} R_{\ell j}^{k}\frac{w_\ell}{\mu_\ell}\right)\times\sum_{\ell=1}^{N}(R_{i\ell}^{k+1}-R_{i\ell}^{k})\frac{w_\ell}{\mu_\ell}\Big]\Big\}+$$

$$\Big\{(c^{k+1}-c^k)\left(1+\frac{1}{2}\sum_{\ell=1}^{N} R_{i\ell}^{k}\frac{w_\ell}{\mu_\ell}\right)\left(1+\frac{1}{2} R_{\ell j}^{k}\frac{w_\ell}{\mu_\ell}\right)\times$$

$$(-10\ b\ \mathrm{sech}^2\ 10(\tau_1-c^k))\Big\},$$

$$\frac{dc^{k+1}}{d\tau_1} = 0.$$

Since $N = 7$, there are basically $7^2 + 1 = 50$ differential equations, which reduce to $7\cdot 8/2 + 1 = 29$ differential equations by the use of the symmetry property

$$(5)\quad R_{ij}^{k+1}(\tau_1) = R_{ji}^{k+1}(\tau_1).$$

While the computations are reduced, the full set of values R_{ij}^{k+1} representing a 7×7 matrix is always available.

Now let the 50-dimensional vector $x^{k+1}(\tau_1)$ have the components

(6) $x_{\ell}^{k+1}(\tau_1) = R_{ij}^{k+1}(\tau_1)$,

for $\ell = 1,2,\ldots,49$ as $i = 1,2,\ldots,7$ and $j = 1,2,\ldots,7$, and

(7) $x_{50}^{k+1}(\tau_1) = c^{k+1}(\tau_1)$.

Since $x^{k+1}(\tau_1)$ is a solution of a system of linear differential equations, we may represent it as the sum of a particular vector solution, $p(\tau_1)$, and a vector solution of the homogeneous sytem, $h(\tau_1)$,

(8) $x^{k+1}(\tau_1) = p(\tau_1) + m\, h\,(\tau_1)$.

The system of differential equations for $p(\tau_1)$ is obtained by substituting the appropriate component of p where ever R^{k+1} or c^{k+1} occurs in (4). We choose the initial conditions $p(0) = 0$. The system of equations for the homogeneous solution is similarly obtained, but of course all terms not involving the $(k+1)^{st}$ approximation are dropped. The initial vector $h(0)$ has all of its components zero except for the last, which is unity. The boundary conditions $R_{ij}^{k+1}(0) = 0$ are identically satisfied. The solutions $p(\tau_1)$ and $h(\tau_1)$ are produced on the interval $0 \leq \tau_1 \leq 1.0$ by numerical integration.

The multiplier m is chosen to minimize the quadratic form,

(9) $$S = \sum_{\ell=1}^{49} \{p_\ell(1) + m\, h_\ell(1) - b_\ell\}^2 ,$$

where the observations are $b_\ell \cong x_\ell^{k+1}(1)$. It is required that

(10) $$\frac{\partial S}{\partial m} = 0,$$

and so the value of m is

(11) $$m = \frac{\sum_{\ell=1}^{49} h_\ell(1)[b_\ell - p_\ell(1)]}{\sum_{\ell=1}^{49} [h_\ell(1)]^2} .$$

The thickness of the lower layer in the new approximation is

(12) $$c^{k+1} = m.$$

The initial approximation required for this successive approximation scheme is produced by numerically integrating the nonlinear system of equations for R using a rough estimate of c. The results of three experiments with initial guesses c = 0.2, 0.8, and 0.0 respectively are given in Table 3. The values of c obtained in the first, second, third and fourth approximations are tabulated.

TABLE 3

SUCCESSIVE APPROXIMATIONS OF c, THE LEVEL OF THE INTERFACE

Approximation	Run 1	Run 2	Run 3
0	0.2	0.8	0.0
1	0.62	0.57	
2	0.5187	0.5024	No
3	0.500089	0.499970	convergence
4	0.499990	0.499991	
True Value	0.5	0.5	0.5

The initial guess of c in Run 1 is 60% too low, and in Run 2, 60% too high. Yet the correct value of c is accurately found in 3 to 4 iterations. The time required for each run is about 2 minutes on the IBM 7044 digital computer, using an Adams-Moulton fourth order integration scheme with a grid size of $\Delta\tau_1 = 0.01$. Each iteration requires the integration of $2 \times 29 = 58$ differential equations with initial values, and the values of $p_\ell(\tau_1)$ and $h_\ell(\tau_1)$ thus produced are stored in the rapid access memory of the computer at each of a hundred and one grid points, $\tau_1 = 0, .01, .02, \ldots, 1.0$. The current approximation of R^k_{ij} is also stored at a hundred and one points.

Run 3 is an unsuccessful experiment because the initial guess for c, i.e., a single layer approximation, is very poor. The solution diverges.

8. NUMERICAL EXPERIMENTS II. DETERMINATION OF T, THE OVERALL OPTICAL THICKNESS

Now let us try to estimate the total optical thickness T of the stratified medium, assuming that we know all of the other parameters of the system. Again we are provided with 49 measurements of $\{b\}$, the intensity of the diffusely reflected radiation in various directions.

The quantity T is the end point of the range of integration, i.e., $0 \leq \tau_1 \leq T$. In order to have a known end point, we define a new independent variable σ,

(1) $$\sigma T = \tau_1 ,$$

so that the integration interval is fixed, $0 \leq \sigma \leq 1$. Then T satisfies the equation, $dT/d\sigma = 0$. Our system of non-linear equations is

$$(2) \qquad \frac{dR_{ij}(\sigma)}{d\sigma} = T \left\{ -\left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R_{ij} + \lambda \left[1 + \frac{1}{2} \sum_{k=1}^{N} R_{ij} \frac{w_k}{\mu_k}\right] \left[1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj} \frac{w_k}{\mu_k}\right] \right\},$$

$$\frac{dT}{d\sigma} = 0,$$

where $\lambda = a + b \tanh 10(\sigma T - c)$.

The solution is subject to the conditions

$$(3) \qquad R_{ij}(0) = 0,$$

$$(4) \qquad \min_{T} \sum_{i=1}^{N} \sum_{j=1}^{N} [R_{ij}(T) - 4\mu_i b_{ij}]^2 .$$

Linear differential equations are obtained in the same manner as before, and we solve a sequence of linear boundary value problems.

Three trials are made to determine the thickness T, with initial guesses $T = 0.9$, 1.5, and 0.5, while the correct value is 1.0. Four iterations yield a value of T which is correct to one part in a hundred thousand, in each of the three experiments. The total computing time is four minutes. The experiment is successful even when the initial guess is only one-half of the true value.

9. NUMERICAL EXPERIMENTS III. DETERMINATION OF THE TWO ALBEDOS AND THE THICKNESS OF THE LOWER LAYER

Given 49 measurements of the diffusely reflected light, we wish to determine the two albedos

$$(1) \qquad \lambda_1 \cong a-b, \qquad \lambda_2 \cong a+b,$$

and the thicknesses of the two layers. We assume that we know the overall thickness $T = 1.0$, and so if the thickness of the lower layer is c, the thickness of the upper layer is given by $T - c$. The unknown parameters are a, b, and c. Since there are three unknowns, we have three homogeneous solutions and of course a particular solution to compute in each iteration of the experiment. Each solution has $28 + 3 = 31$ components, so that there are $4 \times 31 = 124$ linear differential equations being integrated during each stage of the quasilinearization scheme. The three multipliers form the solution of a third order linear algebraic system. They are found by a matrix inversion using a Gaussian elimination method. Table 4 summarizes the results of an experiment which is carried out in about 2 minutes on the IBM 7044. The FORTRAN IV computer programs for all three series of experiments are given in Appendix B.

TABLE 4

SUCCESSIVE APPROXIMATIONS OF λ_1, λ_2, AND c

Approximation	$\lambda_1 = a-b$	$\lambda_2 = a+b$	c
0	0.51	0.69	0.4
1	0.4200	0.6052	0.5038
2	0.399929	0.599995	0.499602
3	0.399938	0.599994	0.499878
True Value	0.4	0.6	0.5

10. DISCUSSION

The approach which is discussed above is readily extended to other inverse problems with different physical situations. The numerical experiments in this chapter make use of many accurate observations of the reflected light while in the next chapter, the effect of errors in the measurements is examined. We note that initial approximations must be good enough to insure convergence. A rational initial estimate may be made from knowledge of the diffuse reflection fields for various homogeneous slabs, as calculated for example by Bellman, Kalaba and Prestrud [14]. Other inverse problems might deal with the transmission function, the source function, the X and Y functions, and the emergence probabilities [18–22].

REFERENCES

1. Hille, E., and R. Phillips, Functional Analysis and Semi-Groups, American Mathematical Society, Providence, R. I., 1957.

2. Frank, Philipp, and Richard Von Mises, Die Differential- und Integralgleichungen, Mary S. Rosenberg, New York, 1943.

3. Ambarzumian, V. A., "Diffuse Reflection of Light by a Foggy Medium," Compt. Rend. Acad. Sci. U.S.S.R., Vol. 38, No. 8, 229-232, 1943.

4. Chandrasekhar, S., Radiative Transfer, Oxford University Press, London, 1950.

5. Bellman, R., and R. Kalaba, "On the Principle of Invariant Imbedding and Propagation through Inhomogeneous Media," Proc. Nat. Acad. Sci. USA, Vol. 42, 629-632, 1956.

6. Bellman, R., and R. Kalaba, "On the Principle of Invariant Imbedding and Diffuse Reflection from Cylindrical Regions," Proc. Nat. Acad. Sci. USA, Vol. 43, 517-520, 1957.

7. Bellman, R., and R. Kalaba, "Invariant Imbedding, Wave Propagations and the WKB Approximation," Proc. Nat. Acad. Sci. USA, Vol. 44, 317-319, 1958.

8. Bellman, R., and R. Kalaba, "Functional Equations, Wave Propagation, and Invariant Imbedding," J. Math. Mech., Vol. 8, 683-704, 1959.

9. Bellman, R., and R. Kalaba, "Transport Theory and Invariant Imbedding," Nuclear Reactor Theory, American Mathematical Society, Providence, R. I., 206-218, 1961.

10. Bellman, R., R. Kalaba, and G. M. Wing, "Invariant Imbedding and Mathematical Physics-I: Particle Processes," J. Math. Phys., Vcl. 280-308, 1960.

11. Wing, G. M., An Introduction to Transport Theory, John Wiley and Sons, Inc., New York, 1962.

12. Kourganoff, V., Basic Methods in Transfer Problems, Oxford University Press, London, 1952.

13. Bellman, R. E., H. H. Kagiwada, R. E. Kalaba, and M. C. Prestrud, Invariant Imbedding and Time-Dependent Transport Processes, American Elsevier Publishing Co., Inc., New York, 1964.

14. Bellman, R., R. Kalaba, and M. Prestrud, Invariant Imbedding and Radiative Transfer in Slabs of Finite Thickness, American Elsevier Publishing Co., Inc., New York, 1963.

15. Shimizu, A., Application of Invariant Imbedding to Penetration Problems of Neutrons and Gamma Rays, NAIG Nuclear Research Laboratory, Research Memo-1, December 1963.

16. Fujita, H., K. Kobayashi, and T. Hyodo, "Backscattering of Gamma Rays from Iron Slabs," Nuclear Science and Engineering, Vol. 19, pp. 437-440, 1964.

17. Minnaert, M., "The Reciprocity Principle in Lunar Photometry," Astrophys. J., Vol. 93, pp. 403-410, 1941.

18. Bellman, R., H. Kagiwada, R. Kalaba, and S. Ueno, A New Derivation of the Integro-Differential Equations for Chandrasekhar's X and Y Functions, The RAND Corporation, RM-4349-ARPA, December 1964.

19. Bellman, R., H. Kagiwada, R. Kalaba, and S. Ueno, Analytical and Computational Techniques for Multiple Scattering in Inhomogeneous Finite Slabs, The RAND Corporation, RM-4438-PR, January 1965.

20. Bellman, R., H. Kagiwada, R. Kalaba, and S. Ueno, Invariant Imbedding and the Computation of Internal Fields for Transport Processes, The RAND Corporation, RM-4472-PR, February 1965.

21. Busbridge, I. W., The Mathematics of Radiative Transfer, Cambridge University Press, 1960.

22. Sobolev, V. V., A Treatise on Radiative Transfer, D. Van Nostrand Company, Princeton, 1963.

BLANK PAGE

CHAPTER THREE

INVERSE PROBLEMS IN RADIATIVE TRANSFER: NOISY OBSERVATIONS

1. INTRODUCTION

The techniques of invariant imbedding and quasilinearization are applied to some inverse problems of radiative transfer through an inhomogeneous slab in which the albedo for single scattering has a parabolic dependence on optical height. The results of many numerical experiments on the effect of the angle of incidence of radiation, errors in observations, and minimax versus least squares criterion are reported. Other experiments are carried out to design an optical medium according to specified requirements. The knowledge gained through this type of numerical experimentation should prove useful in the planning of laboratory or satellite experiments as well as for the reduction of data and the construction of model atmospheres.

2. AN INVERSE PROBLEM

Consider an inhomogeneous, plane-parallel, non-emitting and isotropically scattering atmosphere of finite optical thickness τ_1. Its optical properties depend only on the optical distance τ from the bottom surface. The bottom surface is a completely absorbing boundary, so that no light is reflected from it. See Fig. 1 for a sketch of the physical situation. Parallel rays of light of net flux π per unit area normal to their direction of propagation are incident on the upper surface. The direction is specified by μ_0 $(0 < \mu_0 \leq 1)$, the cosine of the angle measured from the normal to the surface [1,2].

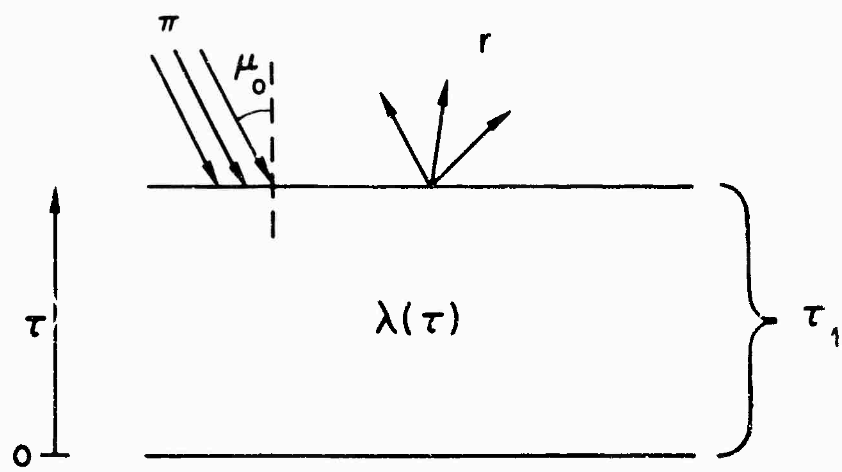


Fig. 1. The physical situation

Let $r(\mu, \mu_0, \tau_1)$ denote the intensity of the diffusely reflected light in the direction μ, and set $R(\mu, \mu_0, \tau_1) = 4\mu r$. Then the function R satisfies the integro-differential equation

$$\text{(1)} \quad \frac{\partial R}{\partial \tau_1} = -\left(\frac{1}{\mu} + \frac{1}{\mu_0}\right) R + \lambda(\tau_1)\left\{1 + \frac{1}{2}\int_0^1 R(\mu, \mu', \tau_1)\frac{d\mu'}{\mu'}\right\} \cdot \left\{1 + \frac{1}{2}\int_0^1 R(\mu', \mu_0, \tau_1)\frac{d\mu'}{\mu'}\right\}$$

with initial condition

(2) $R(\mu, \mu_0, \tau_1) = 0$.

The function $\lambda(\tau_1)$ is the albedo for single scattering.

We wish to consider the inverse problem of estimating the optical properties of the medium as represented by $\lambda(\tau)$ as well as the optical thickness of the slab, based on measurements of the diffusely reflected light.

3. FORMULATION AS A NONLINEAR BOUNDARY VALUE PROBLEM

Let us consider the case in which the albedo may be assumed to have a parabolic form,

(1) $$\lambda(\tau) = \frac{1}{2} + a\tau + b\tau^2 ,$$

where a and b are constants for a particular slab. For example, let us take $a = 2$ and $b = -2$, and we choose the optical thickness $c = 1.0$. The albedo as a function of optical height is shown in Fig. 2.

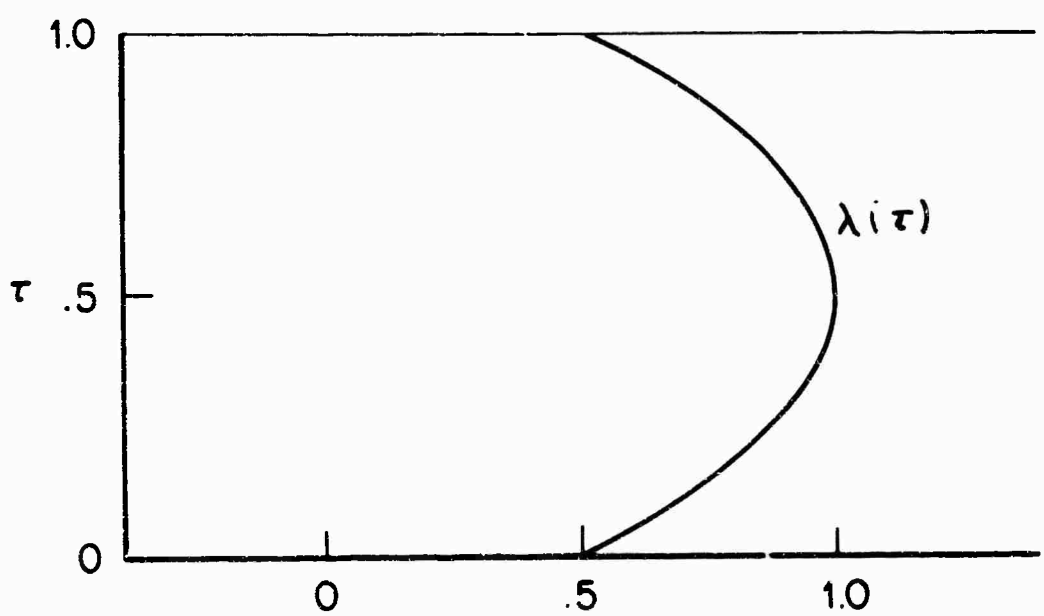


Fig. 2. A parabolic albedo function, $\lambda(\tau) = \frac{1}{2} + 2\tau - 2\tau^2$, for a slab of thickness 1.0

We replace the integro-differential equation by the discrete approximate system obtained by the use of Gaussian quadrature,

$$(2) \quad \frac{dR_{ij}}{d\tau_1} = -\left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R_{ij} + \lambda(\tau_1) \left\{1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik}(\tau_1) \frac{W_k}{\mu_k}\right\} \cdot \left\{1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj}(\tau_1) \frac{W_k}{\mu_k}\right\}.$$

In these equations, $R_{ij}(\tau_1)$ represents $R(\mu_i, \mu_j, \tau_1)$.

We produce "observations" of the diffusely reflected light by choosing $N = 7$, and integrating (2) from $\tau_1 = 0$ to $\tau_1 = 1.0$, and then setting $b_{ij} = \frac{R_{ij}}{4\mu_i}$. Then $\{b_{ij}\}$ is the set of measurements for $\tau_1 = 1$.

Starting with the observations $\{b_{ij}\} \cong \{r_{ij}(c)\}$, we wish to determine the quantities a, b, and the optical thickness c which minimize the expression

$$(3) \qquad S = \sum_{i,j} \{r_{ij}(c) - b_{ij}\}^2 ,$$

where $R_{ij}(\tau_1) = 4\mu_i r_{ij}(\tau_1)$ is the solution of the nonlinear system (2). This inverse problem may be viewed as a nonlinear boundary-value problem.

4. SOLUTION VIA QUASILINEARIZATION

Since the terminal value of the independent variable τ_1 is unknown, we make the following transformation to a new independent variable σ,

(1) $\sigma = \tau_1/c$,

which has initial value 0 and terminal value 1.0. Then the parameters a, b, and the thickness c satisfy the equations

(2) $$\frac{da}{d\sigma} = 0 , \quad \frac{db}{d\sigma} = 0 , \quad \frac{dc}{d\sigma} = 0 .$$

Eqs. (2) are added to the system

(3) $$\frac{dR_{ij}}{d\sigma} = c \left\{- \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R_{ij} + \lambda(\sigma) \left[1 + \frac{1}{2} \sum_{k=1}^{N} R_{ik} \frac{W_k}{\mu_k}\right] \cdot \left[1 + \frac{1}{2} \sum_{k=1}^{N} R_{kj} \frac{W_k}{\mu_k}\right]\right\}$$

where

(4) $$\lambda(\sigma) = \frac{1}{2} + ac\sigma + bc^2\sigma^2 .$$

The application of the technique of quasilinearization [2] yields the linear system for the $(n+1)^{st}$ approximation,

$$\frac{dR_{ij}^{n+1}}{d\sigma} = c^n \left\{- \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R_{ij}^n + \lambda(a^n, b^n, c^n, \sigma) f_i(R^n) f_j(R^n)\right\}$$

$$+ c^n \left\{- \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)(R_{ij}^{n+1} - R_{ij}^n)\right.$$

$$+ \frac{1}{2} \lambda(a^n, b^n, c^n, \sigma) \left[f_i \sum_{k=1}^{N} (R_{kj}^{n+1} - R_{kj}^n) \frac{W_k}{\mu_k}\right.$$

$$\left.\left. + f_j \sum_{\ell=1}^{N} (R_{i\ell}^{n+1} - R_{i\ell}^n) \frac{W_\ell}{\mu_\ell} \right]\right\}$$

$$+ (a^{n+1} - a^n)\, c^n\, (c^n\sigma)\, f_i\, (R^n)\, f_j\, (R^n)$$

$$+ (b^{n+1} - b^n)\, c^n\, (c^n\sigma)^2\, f_i\, (R^n)\, f_j\, (R^n)$$

$$+ (c^{n+1} - c^n) \left\{ -\left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) R^n_{ij} + \lambda(a^n, b^n, c^n, \sigma) f_i(R^n) f_j(R^n) \right.$$

$$\left. + [a^n c^n \sigma + 2b^n (c^n\sigma)^2] \cdot f_i(R^n) f_j(R^n) \right\} , \tag{5}$$

$$\frac{da^{n+1}}{d\sigma} = 0 \ ,$$

$$\frac{db^{n+1}}{d\sigma} = 0 \ ,$$

$$\frac{dc^{n+1}}{d\sigma} = 0 \ ,$$

where

$$\lambda(a^n, b^n, c^n, \sigma) = \frac{1}{2} + a^n(c^n\sigma) + b^n(c^n\sigma)^2 \ ,$$

$$f_i(R^n) = 1 + \frac{1}{2} \sum_{j=1}^{N} R^n_{ij} \frac{W_j}{\mu_j} \ .$$

The solution of Eqs. (5) may be represented in the form

$$R^{n+1}_{ij}(\sigma) = p_{ij}(\sigma) + \sum_{k=1}^{3} c^k h^k_{ij}(\sigma) \ ,$$

$$a^{n+1}(\sigma) = q_1(\sigma) + \sum_{k=1}^{3} c^k w^k_1(\sigma) \ ,$$

$$b^{n+1}(\sigma) = q_2(\sigma) + \sum_{k=1}^{3} c^k w^k_2(\sigma) \ ,$$

$$c^{n+1}(\sigma) = q_3(\sigma) + \sum_{k=1}^{3} c^k w^k_3(\sigma) \ .$$

where the vector P, constituted of elements $p_{ij}(\sigma)$ and $q_\ell(\sigma)$, is a particular solution of (5), and the vectors H^k composed of elements $h^k_{ij}(\sigma)$ and $w^k_\ell(\sigma)$, are three independent solutions of the homogeneous form of (5), for $k = 1, 2, 3$. We choose the initial conditions $P(0)$ identically zero, and $H^k(0)$ having all of its elements zero except for that component which corresponds to w^k, $k = 1, 2, 3$. The choice of initial conditions allows us to identify the multipliers c^k (not to be confused as powers of c) as

$$
\begin{aligned}
a &= a(0) = c^1 , \\
b &= b(0) = c^2 , \qquad (7) \\
c &= c(0) = c^3 ,
\end{aligned}
$$

We seek the three missing initial values (7).

Let us make the conversion from measurements of $r_{ij}(c)$ to measurements of $R_{ij}(1)$ by setting

$$\beta_{ij} = 4\mu_i b_{ij} . \qquad (8)$$

Then we write the expression to be minimized as

$$S = \sum_{i,j} \left\{R^{n+1}_{ij}(1) - \beta_{ij}\right\}^2 . \qquad (9)$$

This expression is a minimum when the following requirements are met:

$$\frac{\partial S}{\partial a} = 0 , \quad \frac{\partial S}{\partial b} = 0 , \quad \frac{\partial S}{\partial c} = 0 . \qquad (10)$$

By means of (7), these conditions are equivalent to

(11) $$\frac{\partial S}{\partial c^1} = 0 , \quad \frac{\partial S}{\partial c^2} = 0 , \quad \frac{\partial S}{\partial c^3} = 0 .$$

We replace $R_{ij}^{n+1}(1)$ in (9) by its representation (6). Then Eqs. (11) lead us to a third order system of linear algebraic equations of the form

(12) $$AX = B ,$$

where the elements of the matrix A and the vector B are, respectively,

(13) $$A_{ij} = \sum_{m,n} h_{mn}^{i}(1)\, h_{mn}^{j}(1) ,$$

(14) $$B_i = \sum_{m,n} h_{mn}^{i}(1)\, [\beta_{mn} - p_{mn}(1)] ,$$

and the solution vector X has as its components the multipliers c^1, c^2, c^3. In this way we obtain the current approximation to the parameters a and b in the albedo function, and the thickness of the slab, c. To begin the calculations, we produce an initial approximation by integrating the system of nonlinear differential equations (3) with $R(0) = 0$, and using estimated values of the parameters. Several iterations of the method are usually sufficient to attain convergence, if convergence takes place at all.

5. NUMERICAL EXPERIMENTS I: MANY ACCURATE OBSERVATIONS

Some of the observations $\{\beta_{ij}\} \cong \{R_{ij}(1)\}$ are plotted in Fig. 3.

Several types of numerical experiments are carried out. In the first class of experiments, 49 perfectly accurate (to about 8 decimal figures) observations are used to determine the quantities a, b, and c. The 49 observations correspond to measurements for 7 outgoing angles for each of 7 incident directions, as listed in Table 1, Chapter II. In one of the trials, the initial approximation is generated with the guesses $a = 2.2$ (+10% in error), $b = -1.8$ (+10% in error), and $c = 1.5$ (+50% in error). After four iterations, our estimates are decidedly better: $a = 1.99895$ (-.005% in error), $b = -1.99824$ (+.014% in error), and $c = 1.004$ (+.04% in error). We repeat the experiment, with one change: our initial estimate of the thickness is 0.5, only one-half of the correct value. This time the solution diverges and the procedure fails.

Fig. 4a illustrates the rapid rate of convergence to the correct solution for the albedo function $\lambda(\tau)$, for the successful trial. The initial approximation is designated in the figure by the numeral 0, the first approximation by 1. The fourth approximation coincides with the true solution. Fig. 4b shows how the initial approximation to the function $R_{ij}(c)$ for incident direction cosine 0.5 deviates from the observed values as indicated by the curve

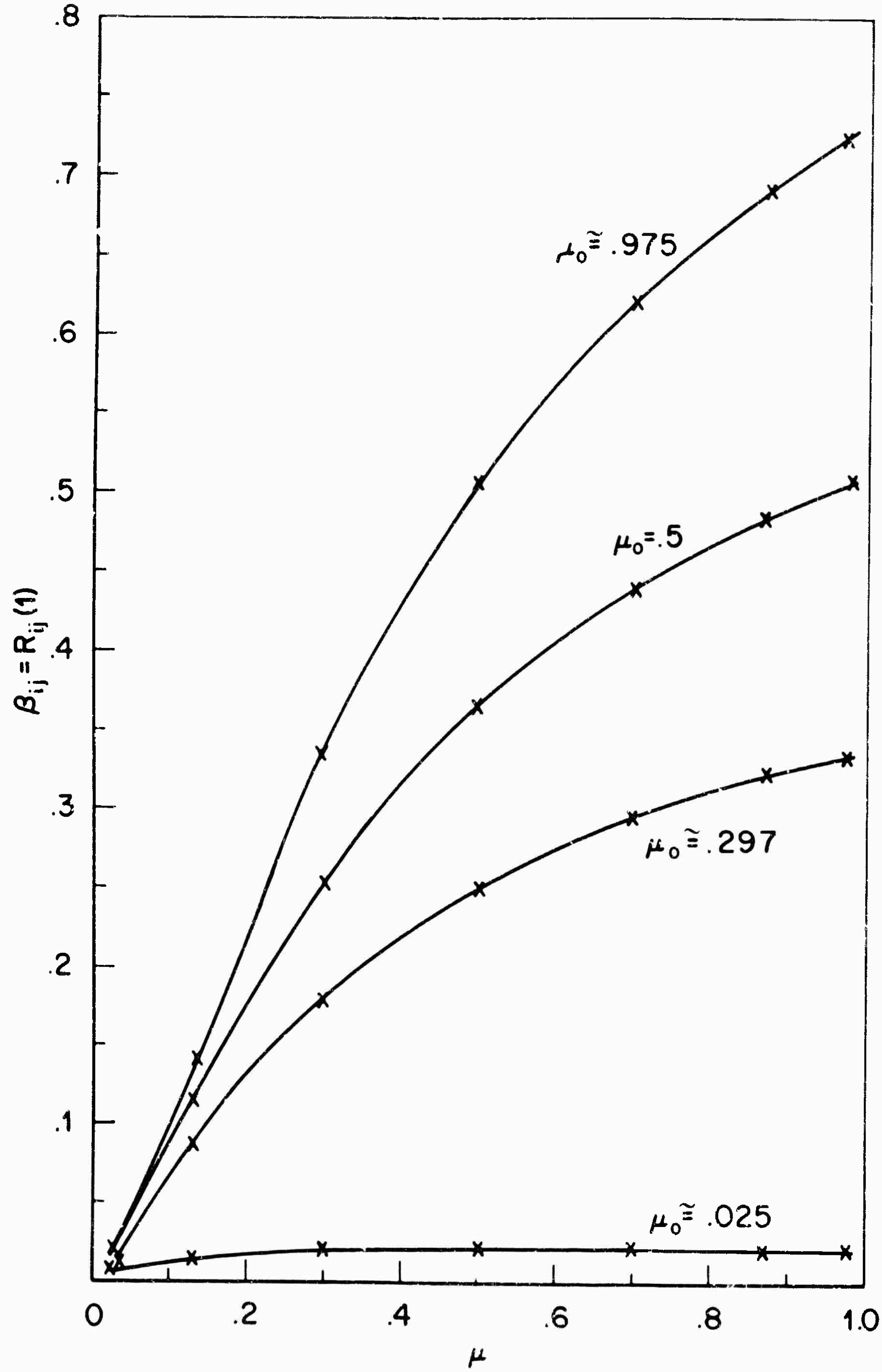
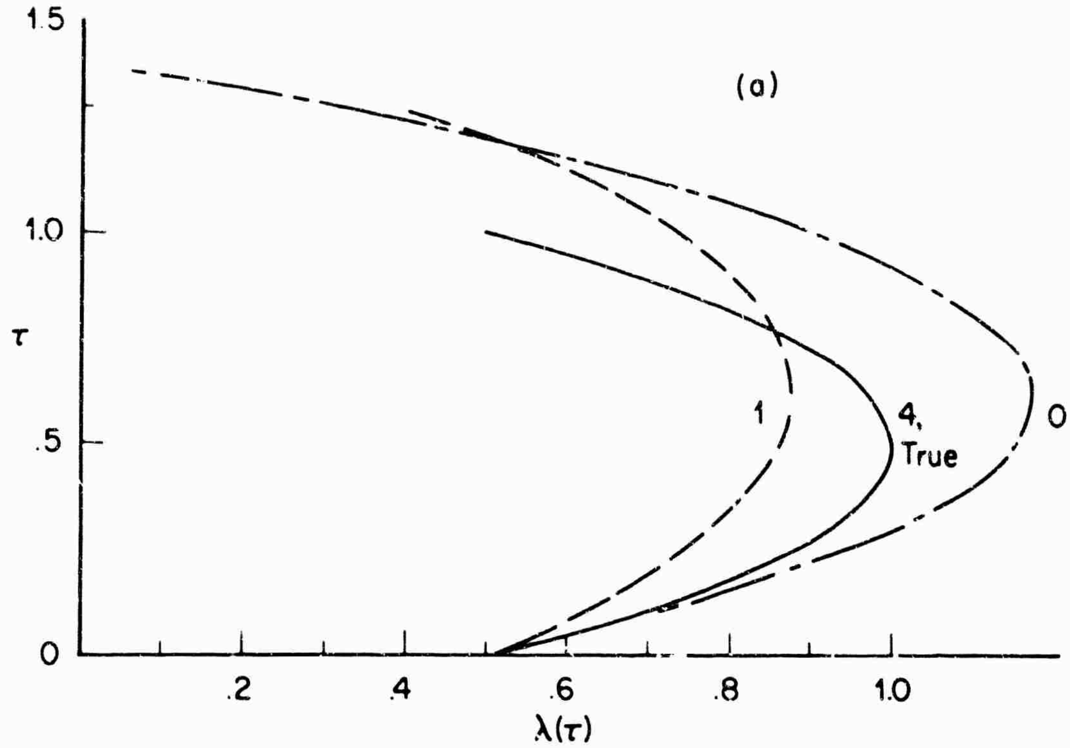


Fig. 3. Some of the observations $\{\beta_{ij}\}$



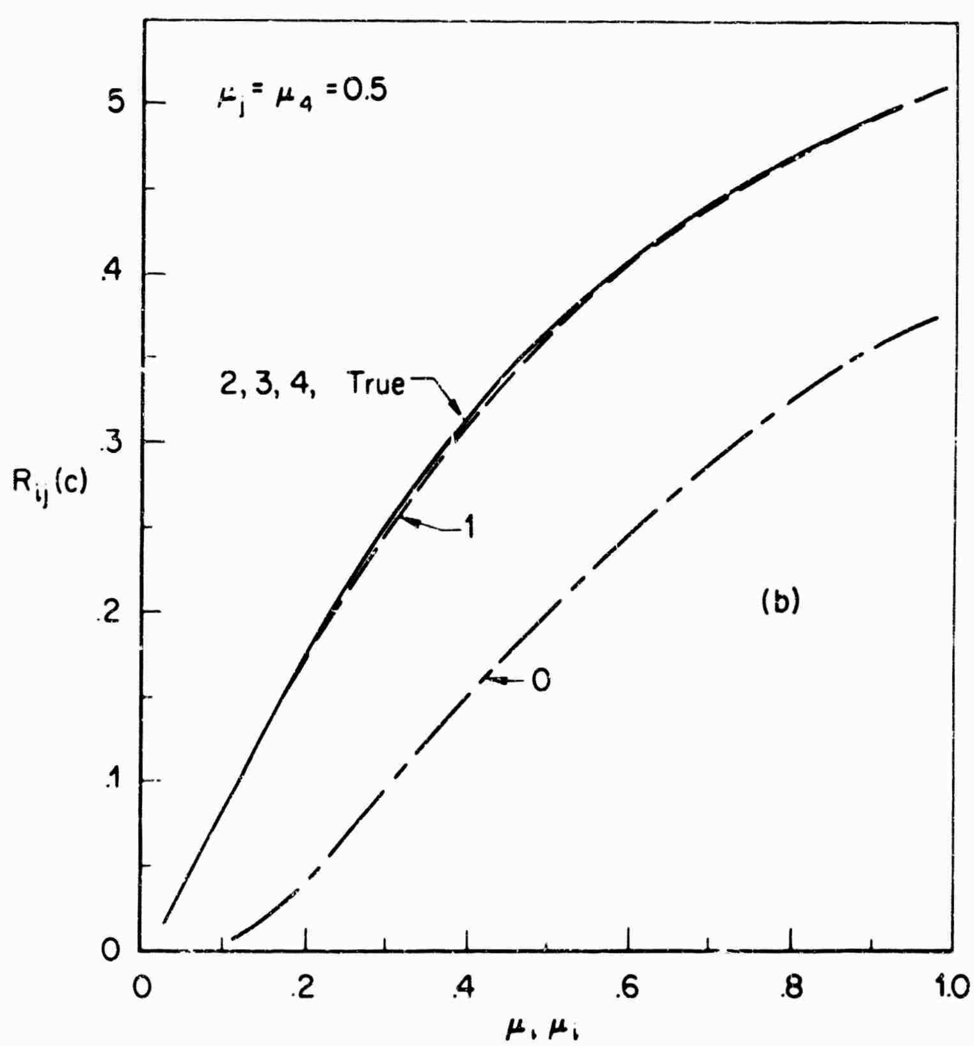


Fig. 4. (a) Successive approximations of the albedo function.
(b) Successive approximations of the function $R_{ij}(c)$.

labelled "True". The first approximation lies very close to the correct values, and the fourth approximation is graphically identical with the correct solution.

6. NUMERICAL EXPERIMENTS II: EFFECT OF ANGLE OF INCIDENCE

In a second series of experiments, the incident angle is held fixed and accurate observations are made of the outgoing radiation in seven directions. The incident direction is varied from one trial to the next in order to study the effect of the position of the source. The initial approximation used in each trial is the same, the correct solution. Due to a possible lack of information in the observations for a given trial, the successive approximations may drift away from the correct solution and converge to another. Several iterations are carried out in each run. The results of the seven runs with each of seven angles are given in Table 1. The incident angle is given in degrees, and the fourth approximations to the constants a, b, and the thickness c are tabulated.

TABLE 1.

NUMERICAL RESULTS WITH DATA FROM
VARIOUS INPUT DIRECTIONS

Trial	Incident Angle	a	b	c
1	88.5°	2.00231	-2.00456	0.999262
2	82.6°	2.00206	-2.00351	0.999361
3	72.7°	2.00032	-2.00048	0.999933
4	60.0°	2.00072	-1.99952	1.00007
5	45.3°	1.99839	-1.99841	1.00021
6	29.5°	2.00029	-2.00040	0.999972
7	13.0°	1.99962	-1.99937	1.00009
Correct values		2.0	-2.0	1.0

Table 1 indicates that the results are very good, no matter what the incident angle is. Examination of the computer output shows that convergence has occurred, in each trial, to about four significant figures. Angles 13° through 72.7° give nearly perfect values of the constants. Angles 82.6° and 88.5°, close to grazing incidence, give values which are only slightly poorer, 0.1% to 0.2% off.

7. NUMERICAL EXPERIMENTS III: EFFECT OF NOISY OBSERVATIONS

In this study, errors of different kinds and amounts are introduced into the observations, and the results of the determination of parameters are compared with the results of Experiments I and II in which no errors were present. Errors are given in percentages with plus or minus signs. The errors in a given trial are either of equal magnitude, or they occur in a Gaussian distribution. Let $t_1, t_2, \ldots, t_7$ be seven true measurements of R. When we speak of noisy observations of $\pm$ 5% equal magnitude errors, we mean that the noisy observations are

$$
(1) \qquad
\begin{aligned}
n_1 &= (1 + .05)t_1 , \\
n_2 &= (1 - .05)t_2 , \\
&\cdots\cdot \\
n_7 &= (1 + .05)t_7 .
\end{aligned}
$$

Let $g_1, g_2, \ldots, g_7$ be seven (signed) Gaussian deviates, with standard deviation unity. Noisy observations with 5% Gaussian distribution of errors are defined to be

$$
(2) \qquad
\begin{aligned}
m_1 &= (1 + .05g_1)t_1 , \\
m_2 &= (1 + .05g_2)t_2 , \\
&\cdots\cdot \\
m_7 &= (1 + .05g_7)t_7 .
\end{aligned}
$$

The results of numerical experiments with noisy observations, with one or seven angles of incidence, are presented in Table 2. Clearly, the accuracy of the estimation of the three constants is in proportion to the

TABLE 2.

NUMERICAL RESULTS WITH ERRORS IN OBSERVATIONS

Incident Angle	± 1% Equal Mag. Error a	b	c	± 2% Equal Mag. Error a	b	c	± 5% Equal Mag. Error a	b	c
88.5°	1.89	−1.80	1.05	1.79	−1.64	1.09	1.5	−1.2	1.3
82.6°	1.99	−1.96	1.013	1.975	−1.93	1.027	1.92	−1.79	1.07
72.7°	1.96	−1.93	1.016	1.92	−1.85	1.03	1.78	−1.61	1.09
60.0°	1.95	−1.91	1.016	1.89	−1.82	1.03	1.69	−1.51	1.10
45.3°	1.94	−1.90	1.016	1.87	−1.79	1.03	1.65	−1.47	1.10
29.5°	1.93	−1.89	1.016	1.86	−1.78	1.03	1.63	−1.44	1.10
13.0°	1.93	−1.89	1.016	1.86	−1.78	1.03	1.62	−1.43	1.10
All 7	1.99	−1.98	1.003	1.96	−1.94	1.009	1.91	−1.85	1.02

Incident Angle	1% Gaussian Error a	b	c	2% Gaussian Error a	b	c
88.5°	1.46	−1.2	1.22	—	—	—
82.6°	1.64	−1.41	1.15	—	—	—
72.7°	1.71	−1.55	1.09	—	—	—
60.0°	1.75	−1.63	1.06	1.54	−1.33	1.13
45.3°	1.77	−1.67	1.05	—	—	—
29.5°	1.78	−1.68	1.05	—	—	—
13.0°	1.79	−1.69	1.04	—	—	—
All 7	1.95	−1.93	1.011	—	—	—

accuracy of the observations. In contrast to the trials with perfect measurements, experiments using noisy observations are more successful when there is an abundance of data, and when the data are limited these experiments show the effect of the incident direction. Errors with Gaussian errors give poorer results, which may be due to the particular set of 7 or 49 Gaussian deviates chosen arbitrarily from a book of random numbers [3].

8. NUMERICAL EXPERIMENTS IV: EFFECT OF CRITERION

This series of experiments is intended to investigate the effect of using a minimax criterion rather than a least squares condition for the determination of the unknown parameters, a, b and c. The condition requires that the constants be chosen to minimize the maximum of the absolute value of the difference between $R_{ij}^{n+1}(1)$ and β_{ij}, where $R_{ij}^{n+1}(1)$ is the solution of (4.5). This is formulated as a linear programming problem in which we have the linear inequalities [4],

$$(1) \qquad \pm \frac{1}{\beta_{ij}} \left\{ p_{ij}(1) + \sum_{k=1}^{3} c^k h_{ij}^k(1) - \beta_{ij} \right\} \leq \epsilon_{ij} \; .$$

$$\epsilon_{ij} \leq \epsilon \; ,$$

where the subscripts take on the values appropriate to the trial under consideration. A standard linear programming code [5] is used to determine the constants c^k, ϵ_{ij}, and the maximum deviation ϵ. Two numerical experiments are carried out, one with $\pm$ 2% equal magnitude errors in the observations, the other with 2% Gaussian errors. The incident angle is 60°. The results are given in Table 3, where we show the values of the two constants in the albedo functions, a and b, the thickness c, and the maximum deviation ϵ, for each approximation. The results for the case where the errors are all of the same relative size are excellent. The trial using Gaussian errors yields constants which are not quite as good, yet these results are surprisingly better than one might expect.

TABLE 3.

NUMERICAL RESULTS USING MINIMAX CRITERION

Type of Errors	Approximation	a	b	c	Maximum Deviation
$\pm$ 2%	0	2.00000	−2.00000	1.00000	———
equal	1	1.99948	−1.99961	1.00001	.0200000
magnitude	2	1.99948	−1.99959	1.00001	.0200000
	3	1.99948	−1.99960	1.00001	.0200000
	0	2.00000	−2.00000	1.00000	———
2%	1	1.76462	−1.67267	1.03357	.0294158
Gaussian	2	1.76265	−1.67487	1.03841	.0293736
	3	1.76279	−1.67484	1.03852	.0293722

9. NUMERICAL EXPERIMENTS V: CONSTRUCTION OF MODEL ATMOSPHERES

Suppose that we desire to construct a model atmosphere with the optical property that whenever light is incident at angles near the normal, the distribution of diffusely reflected light is greatest close to 90° from the normal. We require that the optical thickness c be about 1.0, and the albedo profile is to be parabolic,

$$(1) \qquad \lambda(\tau) = \frac{1}{2} + a\tau + b\tau^2,$$

where the constants a and b are to be suitably chosen. The albedo should not be greater than unity.

The reflection pattern, for an incident angle of 13°, is to have the form indicated by the seven x's in Fig. 5a. The units are given relative to an incident net flux of π. As our initial estimate, we believe that the slab should have thickness one, and that the parameters be $a = 2$, and $b = -2$. Then the albedo has the form given in Fig. 5b by the curve labelled "Initial", the horizontal line at $\tau = 1$ indicating the upper surface c. The reflection function has the form given in Fig. 5a by the dots, whose values are much too low in the region $80^{\circ} - 90^{\circ}$. How should the optical design of this slab be modified for better agreement with the requirements? The answer is not at all obvious.

We carry out a numerical experiment in which a better model is to be found, which makes the sum of the squares of the deviations from the desired values a minimum. The condition is to minimize the sum S,

$$(2) \qquad S = \sum_{i=1}^{7} (d_i - r_{ik})^2 ,$$

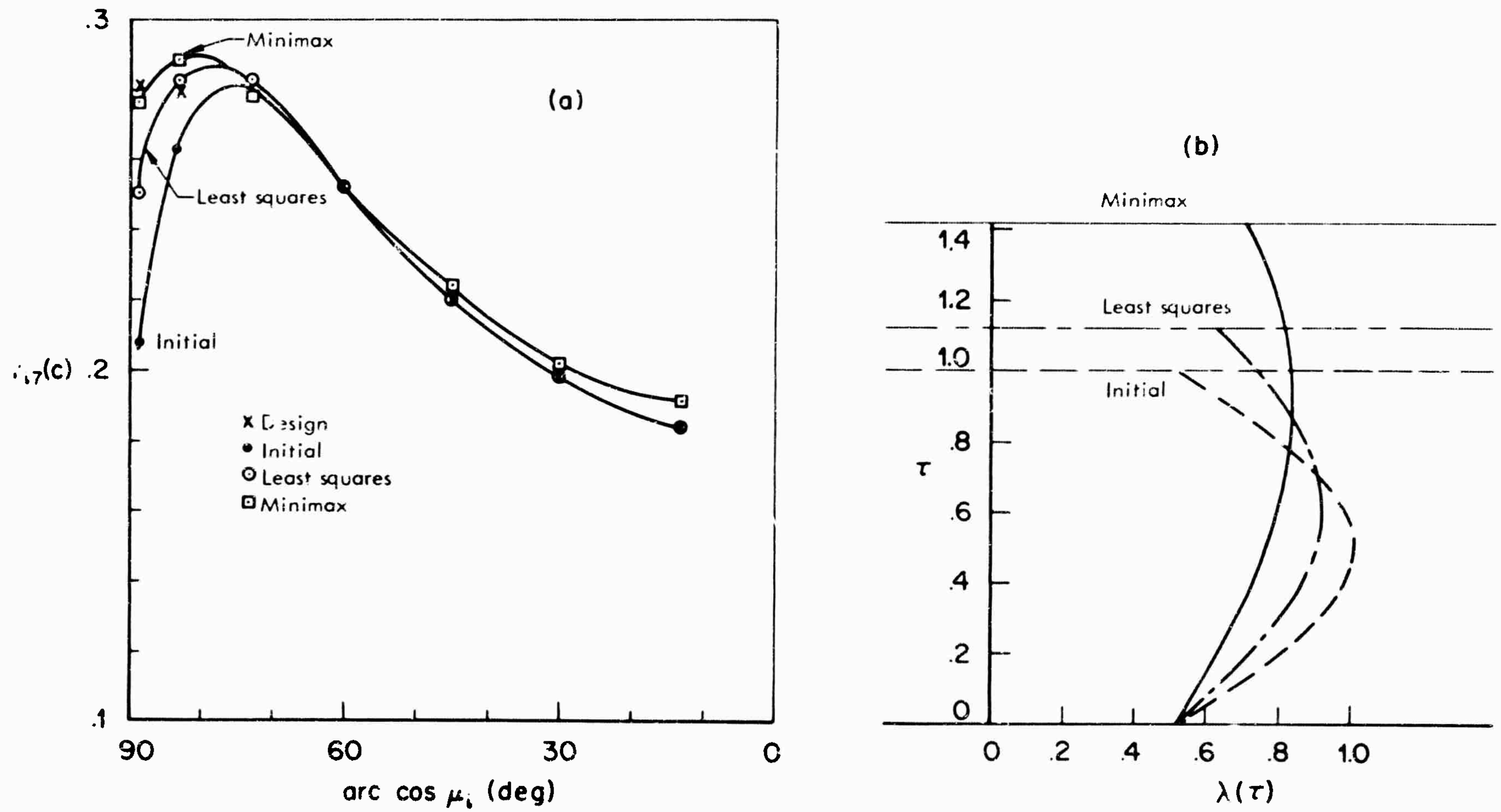


Fig. 5. Several model atmospheres. (a) The diffusely reflected intensity with incident angle 13°.
(b) The albedo function.

where d_i is the desired value of the reflection function for output angle arc cosine μ_i, and r_{ik} is the solution of the differential equations for the r function, and $k = 7$, corresponding to input angle of 13^o. This problem is mathematically the same as the earlier inverse problems of this chapter. The method of solution is also similar, and after five iterations and 3 minutes of computing time, we obtain the solution $a = 1.383$, $b = -1.140$, and $c = 1.117$. The albedo function is shown in Fig. 5b by the curve labelled "Least squares", the reflection function is indicated by the circled dots in Fig. 5a. A smooth curve is drawn between the dots, showing the probable continuous distribution. This curve is in better agreement with the requirements at 83^o and 88.5^o.

We perform another experiment in which the criterion is to minimize the maximum deviation. This condition is given by Eqs. (8.1), where $\beta_{ij} = 4\mu_i d_i$ and $j = 7$. After five iterations, the solution is $a = 0.744$, $b = -0.415$, and optical thickness $c = 1.431$. The albedo has the form represented in Fig. 5b by the curve labelled "Minimax", and the reflection function is indicated by dots within squares in Fig. 5a. The reflection function for this slab is in very good agreement with the requirements.

Other possible approaches to problems of design include dynamic programming and invariant imbedding [6, 7].

REFERENCES

1. Chandrasekhar, S., Radiative Transfer, Dover Publications Inc., New York. 1960.

2. Bellman. R., H. Kagiwada, R. Kalaba, and S. Ueno. Inverse Problems in Radiative Transfer: Layered Media, The RAND Corporation, RM-4281-ARPA, December 1964.

3. The RAND Corporation, A Million Random Digits with 100,000 Normal Deviates, The Free Press, Glencoe, Illinois, 1955.

4. Dantzig, G., Linear Programming and Extensions, Princeton University Press, Princeton, 1963.

5. Clasen, R. J., "RS MSUB - Linear Programming Subroutine, FORTRAN IV Coded," RAND 7044 Library Routine W009, The RAND Corporation, Santa Monica, 1964.

6. Stagg, G. W., and M. Watson, "Dynamic Programming - Transmission Line Design," IEEE International Convention Record, Part 3, pp. 55-61, 1964.

7. Tierney, M. S., P. Waltman and G. M. Wing, "Some Problems in the Optimal Design of Shields and Reflection in Particle and Wave Physics," Sandia Laboratory, Albuquerque New Mexico.

BLANK PAGE

CHAPTER FOUR

INVERSE PROBLEMS IN RADIATIVE TRANSFER: ANISOTROPIC SCATTERING

1. INTRODUCTION

Inverse problems in radiative transfer for a medium with anisotropic scattering [1, 2] may be treated in a manner similar to that used in the isotropic case. We consider a plane parallel slab of finite optical thickness τ_1. For simplicity let us suppose that both the albedo λ and the anisotropic phase function are independent of optical height. Let us take the phase function to be

$$(1) \qquad p(\theta) = 1 + a \cos \theta , \quad \frac{1}{4\pi} \int_{\Omega} p \, d\Omega = 1 ,$$

where θ is the scattering angle, and a is a parameter of the medium and is to be determined on the basis of measurements of the diffusely reflected radiation. An integration over solid angle gives the normalization condition in (1). Let us consider the case in which $a = 1$. This approximately corresponds to the forward phase diagram of Saturn [3]. It may be noted that Horak considers inverse problems in planetary atmospheres in [3].

Parallel rays of light of net flux π per unit area normal to the direction of propagation are incident on the upper surface of the slab. The direction of the rays is characterized by $\mu_0 (0 < \mu_0 \leq 1)$, the cosine of the polar angle measured from the downward normal, and by the azimuth angle φ_0. The phase function may be written as a function of polar and azimuth angles of incidence and scattering, $p(\mu, \varphi; \mu_0, \varphi_0)$. The lower surface of the medium is a perfect absorber.

Let the diffusely reflected intensity in the direction specified by (μ, φ) be $r(\tau_1, \mu, \varphi; \mu_0, \varphi_0)$, where φ is the azimuth angle $(0 \leq \varphi \leq 2\pi)$. Measurements of r, the set $\{b\}$, are made and we wish to determine the value of a for the slab.

2. THE S FUNCTION

Let us define function S, which is related to the diffusely reflected intensity function r, by the formula

$$(1) \qquad r(\tau_1, \mu, \varphi; \mu_0, \varphi_0) = \frac{S(\tau_1, \mu, \varphi; \mu_0, \varphi_0)}{4\mu}$$

We wish to derive a differential-integral equation for the S function by the method of invariant imbedding [2, 4-6].

Let us define another function ρ,

$$(2) \qquad \rho(\tau_1, \mu, \varphi; \mu_0, \varphi_0) = \frac{\mu}{\mu_0} \frac{r(\tau_1, \mu, \varphi; \mu_0, \varphi_0)}{\pi}$$

which is the reflected radiation per unit horizontal area produced by a unit input of radiation on a unit of area in the top surface. Now we add a thin layer of thickness Δ to the top of the slab of thickness τ_1. The ρ function for this slab may be expressed in the form.

$$
\begin{aligned}
&\rho(\tau_1+\Delta,\mu,\varphi;\ \mu_0,\varphi_0) = \\
&\qquad \rho(\tau_1,\mu,\varphi;\ \mu_0,\varphi_0) - \Delta\left(\frac{1}{\mu}+\frac{1}{\mu_0}\right)\rho(\tau_1,\mu,\varphi;\ \mu_0,\varphi_0) \\
&+ \frac{\Delta}{\mu_0}\frac{\lambda}{4\pi}\, p(\mu,\varphi;\ -\mu_0,\varphi_0) \\
&\qquad + \int_0^{2\pi}\int_0^1 \rho(\tau_1,\mu',\varphi';\ \mu_0,\varphi_0)\frac{\Delta}{\mu'}\frac{\lambda}{4\pi}\, p(\mu,\varphi;\ \mu',\varphi')\,d\mu'\,d\varphi' \\
&+ \int_0^{2\pi}\int_0^1 \frac{\Delta}{\mu_0}\frac{\lambda}{4\pi}\, p(-\mu_0',\varphi_0';\ -\mu_0,\varphi_0)\rho(\tau_1,\mu,\varphi;\ \mu_0',\varphi_0')\,d\mu_0'\,d\mu_0' \\
&+ \int_0^{2\pi}\int_0^1 \rho(\tau_1,\mu',\varphi';\ \mu_0,\varphi_0)\frac{\Delta}{\mu'}\frac{\lambda}{4\pi}\, p(-\mu_0',\varphi_0';\ \mu',\varphi')\,d\mu'\,d\varphi' \\
&\qquad \cdot \int_0^{2\pi}\int_0^1 \rho(\tau_1,\mu,\varphi;\ \mu_0',\varphi_0')\,d\mu_0'\,d\varphi_0' + o(\Delta)\ .
\end{aligned}
\tag{3}
$$

The terms on the right hand side of the equation represent the following processes: (1) no interaction in Δ, (2) absorption in Δ of the incoming and outgoing rays, (3) a single scattering in Δ. (4) multiple scattering in the slab of thickness τ_1 followed by an interaction in Δ, (5) interaction in Δ followed by multiple scattering in the slab below. (6) multiple scattering in the slab of thickness τ_1 followed by an interaction in Δ, followed by multiple scattering in the slab of thickness Δ, and (7) $o(\Delta)$ represents other processes which involve Δ^2, or higher powers of Δ.

Now the relation between S and ρ is

$$(4)\qquad \rho(\tau_1, \mu, \varphi; \mu_0, \varphi_0) = \frac{S(\tau_1, \mu, \varphi; \mu_0, \varphi_0)}{4\pi\mu_0}$$

so that when we substitute this expression into (3), we find that S satisfies the equation

$$\begin{aligned}
S(\tau_1+\Delta,\mu,\varphi;\mu_0,\varphi_0) &= S(\tau_1,\mu,\varphi;\mu_0,\varphi_0) \\
&\quad - \Delta\left(\frac{1}{\mu}+\frac{1}{\mu_0}\right)S(\tau_1,\mu,\mu;\mu_0,\varphi_0) \\
&+ \Delta\lambda p(\mu,\varphi;-\mu_0,\varphi_0) \\
&\quad + \frac{\Delta\lambda}{4\pi}\int_0^{2\pi}\int_0^1 S(\tau_1,\mu',\varphi';\mu_0,\varphi_0)p(\mu,\varphi;\mu',\varphi')\frac{d\mu' d\varphi'}{\mu'} \\
(5)\qquad & \\
&+ \frac{\Delta\lambda}{4\pi}\int_0^{2\pi}\int_0^1 p(-\mu_0',\varphi_0';-\mu_0,\varphi_0)S(\tau_1,\mu,\varphi;\mu_0',\varphi_0')\frac{d\mu_0' d\varphi_0'}{\mu_0'} \\
&+ \frac{\Delta\lambda}{(4\pi)^2}\int_0^{2\pi}\int_0^1 S(\tau_1,\mu',\varphi';\mu_0,\varphi_0)p(-\mu_0',\varphi_0';\mu',\varphi')\frac{d\mu' d\varphi'}{\mu'} \\
&\quad \cdot\int_0^{2\pi}\int_0^1 S(\tau_1,\mu,\varphi;\mu_0',\varphi_0')\frac{d\mu_0' d\varphi_0'}{\mu_0'} + o(\Delta)\ .
\end{aligned}$$

We expand the left hand side of Eq. (5) in powers of Δ,

$$S(\tau_1+\Delta,\ \mu,\ \varphi;\ \mu_0,\ \varphi_0) = S(\tau_1,\ \mu,\ \varphi;\ \mu_0,\ \varphi_0) + \Delta\,\frac{\partial S(\tau_1,\mu,\varphi;\ \mu_0,\varphi_0)}{\partial \tau_1} + o(\Delta). \tag{6}$$

We let $\Delta \to 0$ and we obtain the desired integro-differential equation

$$\begin{aligned}
\frac{\partial S(\tau_1,\mu,\varphi;\ \mu_0,\varphi_0)}{\partial\tau_1} + \left(\frac{1}{\mu}+\frac{1}{\mu_0}\right)S &= \lambda\Big\{p(\mu,\varphi;\ -\mu_0,\varphi_0) \\
&+ \frac{1}{4\pi}\int_0^{2\pi}\int_0^1 S(\tau_1,\mu',\varphi';\ \mu_0,\varphi_0)\,p(\mu,\varphi;\ \mu',\varphi')\frac{d\mu'}{\mu'}d\mu' \\
&+ \frac{1}{4\pi}\int_0^{2\pi}\int_0^1 S(\tau_1,\mu,\varphi;\ \mu_0',\varphi_0')\,p(-\mu_0',\varphi_0';\ -\mu_0,\varphi_0)\frac{d\mu_0'}{\mu_0'}d\varphi_0' \\
&+ \frac{1}{(4\pi)^2}\int_0^{2\pi}\int_0^1 S(\tau_1,\mu',\varphi';\ \mu_0,\varphi_0)\,p(-\mu_0',\varphi_0';\ \mu',\varphi')\frac{d\mu'}{\mu'} \\
&\quad \int_0^{2\pi}\int_0^1 S(\tau_1,\mu,\varphi;\ \mu_0',\varphi_0')\frac{d\mu_0'}{\mu_0'}\,d\varphi_0'\Big\}\ .
\end{aligned} \tag{7}$$

A simplification arises if it is assumed that the phase function may be expanded in the Fourier series

$$p(\mu,\ \varphi,\ \mu_0,\ \varphi_0) = \sum_{m=0}^{M} c_m P_m(\cos\theta), \tag{8}$$

where $P_m(x)$ is the Legendre polynomial of degree m. The angular dependence of expansion (8) may be decomposed into polar and azimuth factors by the use of the addition rule of Legendre functions. Then Eq. (8) becomes

$$(9) \qquad p(\mu,\phi,\mu_0,\phi_0) = \sum_{m=0}^{M} (2-\delta_{0m}) \sum_{i=m}^{M} c_i \frac{(i-m)!}{(i+m)!} P_i^m(\mu) P_i^m(\mu_0) \cos m(\phi-\phi_0).$$

The function $P_i^m(x)$ is the associated Legendre function of degree i, order m. Noting the form of this equation, we expand the S function in a similar manner.

$$(10) \qquad S(\tau_1,\mu,\phi,\mu_0,\phi_0) = \sum_{m=0}^{M} S^{(m)}(\tau_1,\mu,\mu_0) \cos m(\phi-\phi_0) \ .$$

Substitution of (10) in (7) leads to the equations for the Fourier components of S,

$$(11) \qquad \frac{\partial S^{(m)}}{\partial \tau_1} + \left(\frac{1}{\mu} + \frac{1}{\mu_0}\right) S^{(m)} = \lambda(2-\delta_{0m}) \sum_{i=m}^{M} (-1)^{i+m} c_i \frac{(i-m)!}{(i+m)!} \psi_i^m(\mu) \psi_i^m(\mu_0)$$

where

$$(12) \qquad \psi_i^m(\mu) = P_i^m(\mu) + \frac{(-1)^{i+m}}{2(2-\delta_{0m})} \int_0^1 S^{(m)}(\tau_1,\mu,\mu') P_i^m(\mu') \frac{d\mu'}{\mu'} \ ,$$

for $m = 0, 1, 2, \ldots, M$. The functions $S^{(m)}(\tau_1,\mu,\mu_0)$ possess the symmetry property

$$(13) \qquad S^{(m)}(\tau_1,\mu,\mu_0) = S^{(m)}(\tau_1,\mu_0,\mu) .$$

The initial conditions are

$$(14) \qquad S^{(m)}(0,\mu,\mu_0) = 0 \ .$$

By the use of Gaussian quadrature on the interval (0.1). the integrals (12) are replaced by sums. Also. the function $S^{(m)}(\tau_1.\mu.\mu_0)$ is replaced by a function of one independent variable. $S_{ij}^{(m)}(\tau_1)$. where the angles are discretized such that $\mu_0 \rightarrow \mu_j$. and $\mu \rightarrow \mu_i$. i.j = 1, 2. .... N. Then we have the approximate system.

$$(15) \qquad \frac{dS_{ij}^{(m)}(\tau_1)}{d\tau_1} + \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)S_{ij}^{(m)} =$$

$$\lambda(2-\delta_{0m}) \sum_{k=m}^{M} (-1)^{k+m} \frac{(k-m)!}{(k+m)!} c_k \psi_{ki}^m \psi_{kj}^m \; .$$

$$(m = 0, 1. \ldots, M; \quad i = 1. 2. \ldots, N; \quad j = 1. 2, \ldots, N) \; ,$$

where

$$(16) \qquad \psi_{k\ell}^m = P_k^m(\mu_\ell) + \frac{(-1)^{k+m}}{2(2-\delta_{0m})} \sum_{j=1}^{N} S_{\ell j}^{(m)} P_k^m(\mu_j) \frac{W_j}{\mu_j} \; .$$

The discrete cosines μ_j are the roots of the shifted Legendre polynominal of degree N. $P_N^*(x)$ and the quantities W_j are the corresponding weights. The initial conditions are

$$(17) \qquad S_{ij}^{(m)}(0) = 0 \; .$$

The solution of this initial value integration problem for a system of ordinary differential equations (15) is approximately equal to the solution of the original integro-differential system.

3. AN INVERSE PROBLEM

Consider the case in which the slab is of thickness $\tau_1 = 0.2$, the albedo is $\lambda = 1$, and we choose $N = 7$ for the quadrature. For the phase function

(1) $$p = 1 + a \cos \theta ,$$

the parameters are

(2) $$M = 1, \ c_0 = 1, \ c_1 = a = 1 .$$

For a numerical experiment, we take Eqs. (2.15) and integrate from $\tau_1 = 0$ with initial conditions (2.17) to $\tau_1 = 0.2$, using an integration grid size of $\Delta\tau_1 = 0.01$. Using (2.10) and (2.1), we produce

(3) $$b_{ijk} \cong r(0.2, \mu_i, \varphi_k; \mu_j, \varphi_0)$$

for $\varphi_0 = 0$, and $\varphi_k = 0^o, 30^o, 60^o, \ldots, 180^o$ as $k = 1, 2, \ldots, 7$. The set $\{b_{ijk}\}$ represents our measurements of the diffuse reflection field, from which we hope to estimate the unknown parameter a. The condition shall be to minimize the sum of squares of deviations,

(4) $$\sum_{i,j,k} [r(0.2, \mu_i, \varphi_k; \mu_j, \varphi_0) - b_{ijk}]^2$$

where the function r is the solution of the Eqs. (2.15) – (2.17) using (2.1) and (2.10). The measurements for $\varphi = 0^o$ and for $\varphi = 180^o$ are shown in Fig. 1, when $\mu_0 = 0.5$, $\varphi_0 = 0$. These data were produced numerically with the

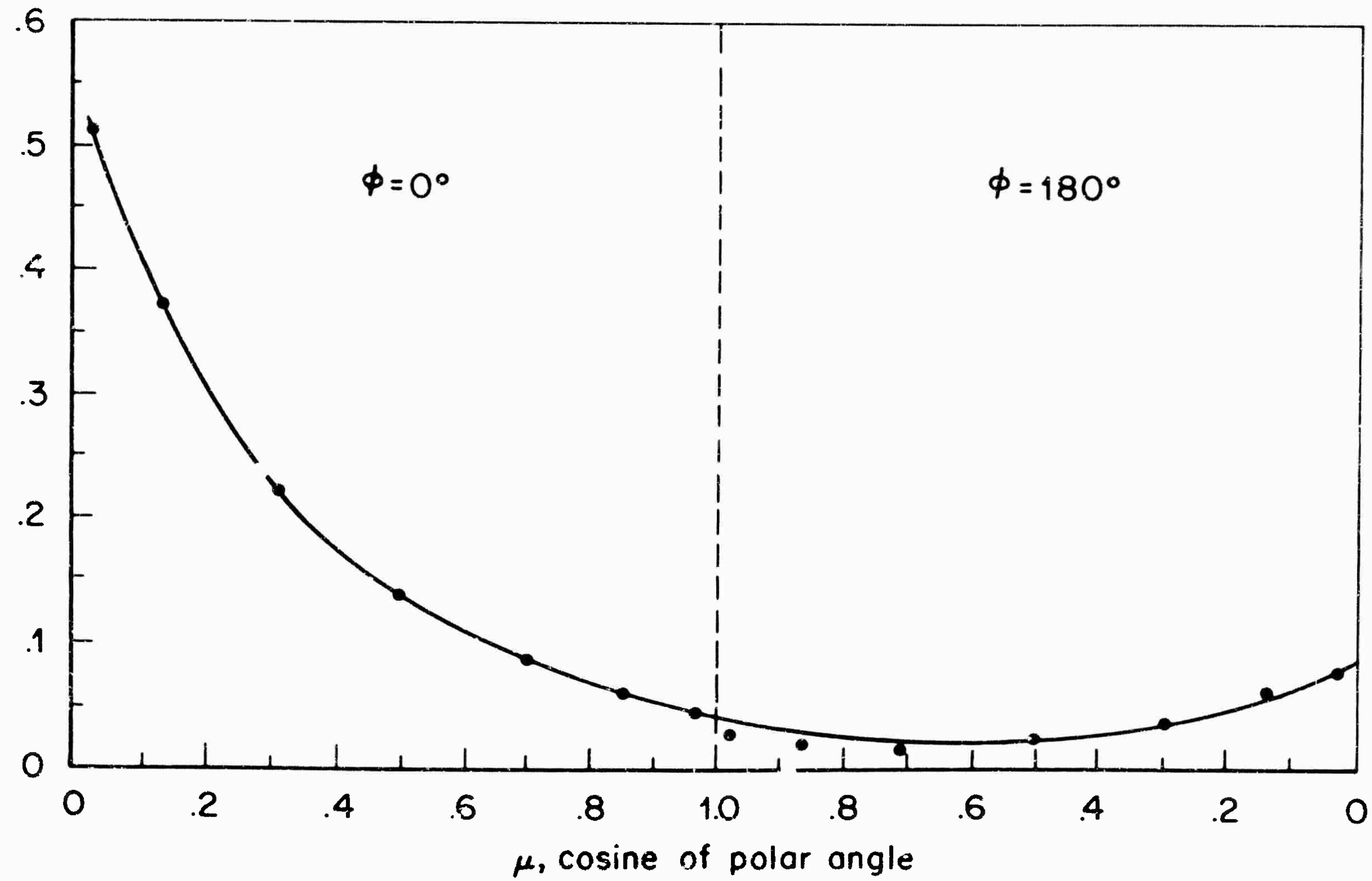


Fig. 1. Fourteen observations of the diffusely reflected intensity $r(\mu,\varphi;\ \mu_0,\varphi_0)$, $\mu_0 = 0.5$, $\varphi_0 = 0^{\circ}$, phase function $p(\theta) = 1 + \cos\theta$.

use of Eqs. (2.1). (2.10) - (2.14). The program for the calculation of the r function is given in Appendix D.

4. METHOD OF SOLUTION

This problem may be solved by successive approximations using quasilinearization [7,8]. Let us write the function $S_{ij}^{(m)}$ as S_{mij}, and similarly $\psi_{ki}^{m} \to \psi_{mki}$, $P_k^m(\mu_\ell) \to P_{mk\ell}$. The linear equations for the $(n+1)^{st}$ approximation are

$$\frac{dS_{mij}^{n+1}}{d\tau_1} = \left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right) S_{mij}^{n}$$

$$+ \lambda(2-\delta_{0m}) \sum_{k=m}^{1} (-1)^{k+m} \frac{(k-m)!}{(k+m)!} c_k^n \psi_{mki} \psi_{mkj}$$

(1) $$+ (S_{mij}^{n+1} - S_{mij}^{n})(-1)\left(\frac{1}{\mu_i} + \frac{1}{\mu_j}\right)$$

$$+ \lambda(2-\delta_{0m}) \sum_{\ell=1}^{N} [(S_{mj\ell}^{n+1} - S_{mj\ell}^{n})\Phi_{mi\ell} + (S_{mi\ell}^{n+1} - S_{mi\ell}^{n+1})\Phi_{mj\ell}]$$

$$+ \lambda(2-\delta_{0m})(c_1^{n+1} - c_1^{n})(-1)^{1+m} \frac{(1-m)!}{(1+m)!} \psi_{m1i}\psi_{m1j}$$

$(m = 0, 1;\ i = 1, 2, \ldots, 7;\ j = 1, 2, \ldots, 7)$,

(2) $$\frac{da^{n+1}}{d\tau_1} = 0 \quad \text{for } a = c_1,$$

where

(3) $$\Phi_{mi\ell} = \sum_{k=m}^{1} \frac{(-1)^{k+m}}{2(2-\delta_{0m})} P_{mkj} \frac{W_j}{\mu_j}(-1)^{k+m} \frac{(k-m)!}{(k+m)!} C_k^n \Psi_{mki}$$

and

(4) $$\Psi_{mk\ell} = P_{mk\ell} + \frac{(-1)^{k(m}}{2(2-\delta_{0m})} \sum_{j=1}^{N} S^n_{m\ell j} P_{mkj} \frac{W_j}{\mu_j} .$$

In these equations

(5) $$a^{n+1} = c_1^{n+1} , \quad a^n = c_1^n , \quad c_0^{n+1} = c_0^n = 1 .$$

The initial conditions are

(6) $$S^{n+1}_{mij} (0) = (0)$$

and the boundary condition is

(7) $$\frac{\partial}{\partial a^{n+1}} \left\{ \sum_{i,j.k} \left[\sum_{m=0}^{1} S^{n+1}_{mij}(0.2)\cos m\varphi_k - 4\mu_i b_{ijk} \right]^2 \right\} = 0 .$$

Let us represent the $(n+1)^{st}$ approximation of S as a linear combination of a particular solution and a homogeneous solution

(8) $$S^{n+1}_{mij}(\tau_1) = p_{mij}(\tau_1) + a\, h_{mij}(\tau_1) .$$

In terms of numerically known quantities,

(9) $$a^{n+1} = \frac{\sum_{i,j,k} (4\mu_i b_{ijk} - p_{oij} - p_{1ij}\cos\varphi k)(h_{oij} + h_{1ij}\cos\varphi k)}{\sum_{i,j,k} [h_{oij} + h_{1ij}\cos\varphi k]^2}$$

where the functions p and h are evaluated at $\tau_1 = 0.2$, and the initial conditions for p and h are suitably chosen.

By making use of the symmetry property of S we need consider, not a system of $2N^2$ equations, but only $2N(N+1)/2 = N(N+1)$ equations. For $N = 7$, this means that $7.8 = 56$ equations define the particular solution, and another 56 define the homogeneous solution. Twenty-one integration grid points cover the range $0 \leq \tau_1 \leq 0.2$, with $\Delta\tau_1 = 0.01$. The storage requirements are 21 x 56 for the p solution, 21 x 56 for the h solution and 21 x 56 for S^n_{mij}. This problem is certainly feasible for numerical solution with the IBM 7044 or the 7090. Numerical experiments will be carried out in the near future. Such studies should prove useful in the planning and analysis of investigations of planetary atmospheres [3,9-18], stellar radiation in the galaxy [19], and radiation fields in the sea [20-22].

REFERENCES

1. Chandrasekhar, S., *Radiative Transfer*, Dover Publications, Inc., New York, 1960.

2. Bellman, R., R. Kalaba, and S. Ueno, "Invariant Imbedding and Diffuse Reflection from a Two-Dimensional Flat Layer," *Icarus*, Vol. 1, pp. 297-303, 1963.

3. Horak, Henry G., "Diffuse Reflection by Planetary Atmospheres," *Ap. J.*, Vol. 112, 445-463, 1950.

4. Bellman, R., H. Kagiwada, R. Kalaba, and M. C. Prestrud, *Invariant Imbedding and Time-Dependent Transport Processes*, American Elsevier Publishing Co., New York, 1964.

5. Bellman, R., and R. Kalaba, "Transport Theory and Invariant Imbedding," *Nuclear Reactor Theory*, Amer. Math. Soc., Providence, R.I., pp. 206-218, 1961.

6. Bellman, R., R. Kalaba, and G. M. Wing, "Invariant Imbedding and Mathematical Physics - I: Particle Processes," *J. Math. Phys.*, Vol. 1, pp. 280-308, 1960.

7. Bellman, R., and R. Kalaba, *Quasilinearization and Boundary Value Problems*, American Elsevier Publishing Co., New York, 1965.

8. Bellman, R., H. Kagiwada, R. Kalaba, and S. Ueno, *Inverse Problems in Radiative Transfer: Layered Media*, The RAND Corporation, RM-4281-ARPA, December 1964.

9. Dave J. V., "Meaning of Successive Iteration of the Auxiliary Equation in the Theory of Radiative Transfer," *Astrophys. J.*, Vol. 140, No. 3, pp. 1292-1302, 1964.

10. Deirmendjian, D., "A Water Cloud Interpretation of Venus' Microwave Spectrum," *Icarus*, Vol. 3, No. 2, pp. 109-120, 1964.

11. Schilling, G. F., "Extreme Model Atmosphere of Mars," *Memoires Soc. R. Sc. Liege*, cinquieme serie, tome VII pp. 448-451, 1962.

12. Van de Hulst, H. C., and W. M. Irvine, "General Report on Radiation Transfer in Planets: Scattering in Model Planetary Atmospheres," *Memoires Soc. R. Sc. Liege*, cinquieme serie, tome VII, pp. 78-98, 1962.

13. Sekera, Z. and J. V. Dave. "Diffuse Transmission of Solar Ultraviolet Radiation in the Presence of Ozone." Astrophys. Journal. Vol. 133. pp. 210 227. 1961.

14. Lenoble. J. and Z. Sekera. "Equation of Radiative Transfer in a Planetary Spherical Atmosphere." Proc. Nat. Acad. Sci.. Vol. 47. No. 3. pp. 372 378. 1961.

15. Van de Hulst. H. C., "Scattering in a Planetary Atmosphere." Astrophys. Journal. Vol. 107. pp. 220-2?5. 1948.

16. Chu. C. M., J. A. Leacock. J. C. Chen and S. W. Churchill. "Numerical Solutions for Multiple. Anisotropic Scattering." Electromagnetic Scattering. edited by M. Kerker. Pergamon Press, Oxford. 1963. pp. 567-582.

17. Malkevich. M. S., "Angular and Spectral Distribution of Radiation R flected by the Earth into Space." Planet. Space Sci., Vol. 11, pp. 681-699. 1963.

18. Wernik. A.. "Extinction of the Night Sky Light in an Anisotropically Scattering Atmosphere." Acta Astronomica. Vol. 12. No. 2, pp. 102-121, 1962.

19. Henyey, L. G., and J. L. Greenstein. "Diffuse Radiation in the Galaxy." Astrophys. Journal. Vol. 93, pp. 70-83. 1941.

20. Preisendorfer, R. W., "Application of Radiative Transfer Theory to Light Measurements in the Sea." L'Institute Geographique National. Paris. International Union of Geodesy and Geophysics. Monographie No. 10. June 1961.

21. Tyler, J. E., "Radiance Distribution as a Function of Depth in an Underwater Environment." Bulletin of the Scripps Institution of Oceanography. Vol. 7. No. 5. pp. 363-412. 1960.

22. Tyler, J. E., and A. Shaules. "Irradiance on a Flat Object Underwater," Applied Optics. Vol. 3. pp. 105-110. 1964.

CHAPTER FIVE

AN INVERSE PROBLEM IN NEUTRON TRANSPORT THEORY

1. INTRODUCTION

The theory of neutron transport and the theory of radiative transfer [1] are devoted to problems of determining the properties of radiation fields produced by given sources in a given medium. Inverse problems in transport theory are those in which we seek to determine the properties of the medium, given those of the indicent radiation and the radiation fields [2-4].

In this chapter, we study inverse problems in transport theory from the point of view of dynamic programming [5]. Our aim is to produce a feasible computational method for estimating the properties of the mdium based upon measurements of radiation fields within the medium. The invariant imbedding approach to transport theory is sketched in Ref. 6.

For ease of exposition we consider a one-dimensional transport process. The method described here can be generalized to the vector-matrix case, and thus to the slab geometry with anisotropic scattering, to wave propagation [7], and to transmission lines.

2. FORMULATION

Consider the one dimensional medium shown in Fig. 1.

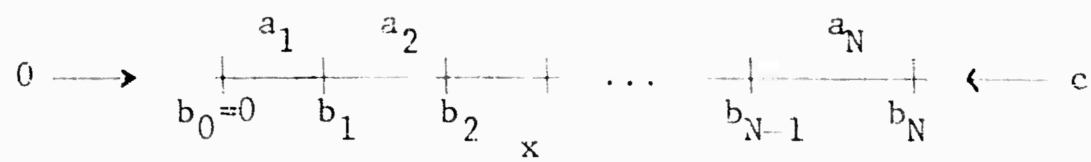


Fig. 1. A one-dimensional transport process.

It corsists of N homogeneous sections (b_i, b_{i+1}), $i = 0, 1, 2, \ldots, N-1$. When a neutron travels through a distance Δ in the i^{th} section, there is probability $a_i\Delta$ that it will interact with the medium. The result of an interaction is that the original neutron is absorbed and two daughter neutrons appear, one traveling in each direction. Suppose that c neutrons per unit time are incident from the right and zero neutrons per unit time from the left. We denote the average number of particles per unit time passing the point x and moving to the right by $u(x)$ and the same quantity for the leftward moving particles by $v(x)$. Suppose that measurements on the internal intensities are made at various points $x = x_i \neq b_j$; e.g.,

(1) $$u(x_i) \cong w_i, \quad i = 1, 2, \ldots, M.$$

Our aim is to estimate the characteristics of the medium, the quantities a_i, $i = 1, 2, \ldots, N$, on the basis of these observations.

As is shown in Ref. 6, the internal intensities satisfy the differential equations

(2)
$$\dot{u} = a_i v ,$$
$$-\dot{v} = a_i u ,$$

(3) $$b_{i-1} \leq x \leq b_i , \quad i = 1, 2, \ldots, N ,$$

where the dot indicates differentiation with respect to x. The analytical solution is of no import, since we wish to consider this as a prototype of more complex processes for which a computational treatment in mandatory. In addition, $u(x)$ and $v(x)$ are continuous at the interfaces

(4) $$u(b_i - 0) = u(b_i + 0)$$

(5) $$v(b_i - 0) = v(b_i + 0) , \quad i = 1, 2, \ldots, N-1 ,$$

and

(6) $$u(0) = 0$$

(7) $$v(b_N) = c .$$

We wish to select the N constants $a_1, a_2, \ldots, a_N$, so as to minimize the sum of the squares of the deviations S,

(8) $$S = \sum_{i=1}^{M} \{u(x_i) - w_i\}^2 .$$

3. DYNAMIC PROGRAMMING

Let us suppose that the functions u and v are subject to the conditions of Section 2 and

(1) $\quad u(b_K) = c_1$

(2) $\quad v(b_K) = c_2$.

In addition we write

(3) $$f_K(c_1, c_2) = \min \sum_{i=1}^{M_K} \{u(x_i) - w_i\}^2 ,$$

where the minimization is over the absorption coefficients $a_1, a_2, \ldots, a_K$. The number of observations on the first K intervals is M_K. We view K as a parameter taking on the values 1, 2, ..., and c_1 and c_2 are also viewed as variables. Then we write

(4) $$f_1(c_1, c_2) = \sum_{i=1}^{M_1} \{u(x_i) - w_i\}^2 ,$$

where

(5) $\quad u(b_1) = c_1$

(6) $\quad v(b_1) = c_2$

(7) $\quad \dot{u} = a_1 v$

(8) $\quad -\dot{v} = a_1 u$.

The absorption coefficient a_1 is chosen so that

(9) $u(0) = 0$.

In addition, the principle of optimality yields the relationship

(10) $$f_{K+1}(c_1, c_2) = \min_{a_{K+1}} \left\{d_{K+1} + f_K(c_1', c_2')\right\},$$
$$K = 1, 2, \ldots .$$

(11) $$d_{K+1} = \sum_i \{u(x_i) - w_i\}^2 .$$

with i ranging over integer values for which

(12) $$b_K < x_i < b_{K+1} ,$$

and

(13) $$\dot{u} = a_{K+1} v , \quad u(b_{K+1}) = c_1$$

(14) $$-\dot{v} = a_{K+1} u , \quad v(b_{K+1}) = c_2 .$$

In addition we have put

(15) $$c_1' = u(b_K)$$

(16) $$c_2' = v(b_K) .$$

In the usual manner of dynamic programming this leads to a computational scheme for computing the sequence of functions of two variables $f_1(c_1, c_2)$, $f_2(c_1, c_2)$, ..., and in principle solves our estimation problem. In the event that we do not wish to require that $u(0) = 0$, we may determine the function $f_1(c_1, c_2)$ this way:

$$(17) \qquad f_1(c_1, c_2) = \min_{a_1} [\lambda\, u^2(0) + \sum_{i=1}^{M_1} \{u(x_i) - w_i\}^2] \, ,$$

where λ is a suitably large parameter.

4. AN APPROXIMATE THEORY

While the original physical problem is a two-dimensional problem, it may be well-represented as a one-dimensional problem. Suppose that there are K segments of the medium and that the input is $v(b_K) = c$. The absorption coefficients $a_1, a_2, \ldots, a_K$ should be chosen to secure a minimum sum of squares of deviations from the measurements. Having picked the absorption coefficients, we may calculate the reflection coefficient $r(v(b_K))$ for this segmented medium. At the end b_K, the function u is essentially determined by v and $r(v)$, $u(b_K) = v(b_K)\, r(v(b_K))$. The single variable $v(b_K) = c$ may then suffice to specify the state at the right end of the K^{th} segment.

Let us define the function $g_N(c)$

(1) $\qquad g_N(c) =$ the smallest sum of squares of deviations on the first N segments when the input is c,

and the function $R_N(c)$,

(2) $R_N(c)$ = the reflection coefficient that results when the optimal set of absorption coefficients is used on the first N segments, the input being $c = v(b_N)$.

The function $g_{N+1}(c)$ satisfies the inequality

$$(3) \qquad g_{N+1}(c) \le \min_a \left\{ d_{N+1} + g_N(c') \right\} ,$$

where

$$(4) \qquad d_{N+1} = \sum_i \left[u(x_i) - w_i\right]^2 , \quad b_N < x_i < b_{N+1} ,$$

and

$$(5) \qquad \begin{aligned} \dot{u} &= a\,v, \quad v(b_{N+1}) = c , \\ -\dot{v} &= a\,u, \quad v(b_N)R_N(v(b_N)) = u(b_N) , \end{aligned}$$

and

$$(6) \qquad c' = v(b_N) .$$

We do not have recurrence relations for the sequences of functions $g_N(c)$ and $R_N(c)$. We replace Eqs. (3), (4), and (5) by an approximate set, where instead of $g_N(c)$ we introduce the sequence $f_N(c)$, and instead of $R_N(c)$ we introduce $r_N(c)$. In our approximate theory we produce $f_{N+1}(c)$ from the recurrence formula

$$(7) \qquad f_{N+1}(c) = \min_a \left\{ d_{N+1} + f_N(c') \right\} ,$$

where

(8) $$d_{N+1} = \sum_i [u(x_i) - w_i]^2 \, , \quad b_N < x_i < b_{N+1} \, .$$

and

$$c' = v(b_N) \, .$$

The following boundary value problem,

(9) $$\begin{aligned} \dot{u} &= a\, v, \quad v(b_{N+1}) = c, \\ -\dot{v} &= a\, u, \quad v(b_N)\, r_N(v(b_N)) = u(b_N) \, . \end{aligned}$$

must be satisfied. The quantity

(10) $$r_{N+1}(c) = r(b_{N+1})$$

is obtained as the solution of the initial value problem

(11) $$\dot{r} = a(1 + r^2), \quad r(b_N) = r_N(c') \, .$$

For $N = 1$ we define

(12) $$f_1(c) = \min_a \sum_i \alpha \, [u(x_i) - w_i]^2 \, ,$$

where the summation is over indices for which

(13) $$0 < x_i < b_1 \, ,$$

and α is a weighting constant. Also we have

(14) $$\begin{aligned} \dot{u} &= a\, v, \quad u(0) = 0 \, , \\ -\dot{v} &= a\, u, \quad v(b_1) = c \, , \end{aligned}$$

We define

$$(15) \qquad r_1(c) = r(b_1)$$

where

$$(16) \qquad \begin{aligned} \dot{r} &= a(1 + r^2) \, , \quad 0 \le x \le b_1 \, . \\ r(0) &= 0 \, . \end{aligned}$$

The purpose of introducing the weight $\alpha \ge 1$ is to insure a good fit over the first segment.

Assuming that a unique minimizing solution exists, we can show that the results of our approximate theory are exact, if the observations w_i are perfectly accurate. We reason inductively. For the one segment process, there exists an input c_1^* for which $f_1(c_1^*) = 0$ by Eq. (12), and the reflection coefficient is $r_1(c_1^*)$. We assume that there exists an input to the medium of N segments, c_N^*, such that $f_N(c_N^*) = 0$, and that the reflection coefficient for this medium is $r_N(c_N^*)$. For the medium of $N+1$ segments, there is an input c_{N+1}^* such that $d_{N+1} = 0$, and the input (to the left) at b_N which satisfies condition (9) is $v(b_N) = c_N^*$. Therefore $f_{N+1}(c_{N+1}^*) = 0$, and the solution is exact.

In this manner we have reduced the original multidimensional optimization process to a sequence of one-dimensional processes.

5. A FURTHER REDUCTION

The solving of the nonlinear boundary value problem of Eq. (4.9) can be a source of difficulty. To aid in this process we note that we can write

$$(1)\qquad v(b_N) = c\ T + u(b_N)R \ ,$$

which follows simply from one of Chandrasekhar's invariance principles [1]. The transmission coefficient T and the reflection coefficient R of the $(N+1)$st segment are calculated from the solutions of the initial-value problems [6]

$$(2)\qquad \dot{r} = a(1 + r^2)\ ,\ r(0) = 0$$

$$(3)\qquad \dot{t} = a\ r\ t,\ t(0) = 1\ ,$$

and

$$(4)\qquad R = r(b_{N+1} - b_N)$$

$$(5)\qquad T = t(b_{N+1} - b_N)\ .$$

In this way the second condition in Eq. (4.9) becomes

$$(6)\qquad r_N(v)v = (v - c\ T)/R\ ,$$

a nonlinear equation for $v = v(b_N)$

6. COMPUTATIONAL PROCEDURE

The calculation of f_{N+1} for a given value of the parameter c may proceed as follows. We take a value of the coefficient a, and we produce numerically the reflection and transmission coefficients, R and T. Assuming we can solve Eq. (5.6) for $v(b_N)$, we go on to solve the linear two-point boundary-value problem of Section 4 by producing numerically two independent solutions of these homogeneous equations and determining constants so that the boundary conditions are met. Then the sum of squares of deviations d is computed, and the cost $\{d + f_N(v(b_N))\}$ is evaluated. We go through these steps for all the admissible choices of a, and the costs are compared. The value of a which makes the cost a minimum is the choice for the $(N+1)^{st}$ slab. The whole procedure is repeated for the range of values of c and of N.

It may be noted that in the calculation of the reflection coefficient r_{N+1}, the initial condition r_N is known only computationally on a grid of values of the argument. Experiments are needed to determine the required fineness of grid to achieve the required accuracy.

It is possible to derive recurrence relations for $f_N'(c)$ and $r_N'(c)$, and these can be employed in a variety of ways to improve the accuracy of the method. Numerical experimentation would have to be carried out to obtain reliable estimates of running times and accuracies [9]. The method proposed here can be extended to treat the case where the interface points are not known, though the computational effort will be greatly increased.

Experience with many similar problems leads us to believe that the proposed procedure is perfectly feasible [8, 9].

7. COMPUTATIONAL RESULTS

Production of observations. We consider a homogeneous rod of unit length with absorption coefficient $a = 0.5$. We produce the internal fluxes to the right and to the left due to a unit input flux to the left at the end $x = 1$, and no input at the other end, $x = 0$. To do this, we use the fact that the quantity $v(1)$ is the reflection coefficient for the slab, which is $\tan a$ [6]. We integrate the transport equations with the initial values $u(1) = 1$, $v(1) = \tan a$, from $x = 1$ to $x = 0$. This procedure yields $u(x)$ and $v(x)$ throughout the rod.

Two-dimensional dynamic programming procedure for the determination of the absorption coefficients. The rod is divided up into 10 homogeneous sections of equal length. From the set of exact measurements, $w_j \approx u(x_j)$, we wish to determine the set of optimizing parameters a_N in each section. The correct solution is $a_N = 0.5$ for $N = 1, 2, \ldots, 10$.

In stage one of the multi-stage decision process, the rod is considered to consist of one segment extending from $x = 0$ to $x = 0.1$. If $c_1 = u(0.1)$, $c_2 = v(0.1)$, we choose the coefficient which makes $u(0) = 0$,

(1) $$a = \frac{1}{0.1} \text{ arc tan } \frac{c_1}{c_2} .$$

regardless of the measurements in this segment. The minimum cost is $f_1(c_1, c_2) = \sum_i [u(x_i) - w_i]^2$, where $u(x_i) = \sin a\, x_i$, $0 < x_i < 0.1$. This calculation is carried out for each value of c_1 and c_2.

The computations for the other stages $N = 2, 3, 4, \ldots$, may be best indicated by the following outline:

TWO-DIMENSIONAL DYNAMIC PROGRAMMING CALCULATIONS

For each stage $N = 2, 3, 4, \ldots$,

1. Print N
2. For each c_1
    1. For each c_2
        1. For each a
            1. Integrate to produce $c_1' = u(b_{N-1})$, $c_2' = v(b_{N-1})$,

               $$\begin{cases} \dot{u} = a\, v, & u(b_N) = c_1 \\ -\dot{v} = a\, u, & v(b_N) = c_2 \end{cases}$$

            2. Compute $d = \sum_i [u(x_i) - w_i]^2$
            3. Find $f_{N-1}(c_1', c_2')$ by interpolation
            4. Set $S(a) = d + f_{N-1}(c_1', c_2')$
        2. Search for $f_N(c_1, c_2) = \underset{a}{\text{Min}}\, \{S(a)\}$
        3. Print c_1, c_2, a_N, c_1', c_2', $f_N(c_1, c_2)$
3. For each c_1
    1. For each c_2
        1. Shift $f_N(c_1, c_2) \longrightarrow f_{N-1}(c_1, c_2)$

There are four levels of computation: the stage N, the state c_1, the state c_2, the parameter a. The large brackets over the steps which must be carried out at each level. By the statement "shift $f_N \to f_{N-1}$", we represent the discarding of the costs for stage $N-1$, and the replacement of f_{N-1} by the just computed costs for stage N, in readiness for the next stage. This saving in storage is allowed by the recurrence formula linking the current cost with the cost of only the previous stage. The interpolation may be carried out by the use of a linear formula in two dimensions, c_1 and c_2. The print-out value of a is, of course, the optimal value.

In our numerical trial, we execute the algorithm for three stages only, the rod then extending from $x = 0$ to $x = 0.3$. The exact observations are

$$
\begin{aligned}
u(0.02) &= 0.11394757 \times 10^{-1} \\
u(0.05) &= .28484388 \times 10^{-1} \\
u(0.08) &= .45567610 \times 10^{-1} \\
u(0.12) &= .68328626 \times 10^{-1} \\
u(0.15) &= .85381951 \times 10^{-1} \\
u(0.18) &= .10241607 \times 10^{0} \\
u(0.22) &= .12509171 \times 10^{0} \\
u(0.25) &= .14206610 \times 10^{0} \\
u(0.28) &= .15900853 \times 10^{0}.
\end{aligned}
$$

The range of N is 1 to 3, the section interfaces lying at $x = 0.1, 0.2, 0.3$. The range of c_1 is 0.00 (0.01) 0.20, 21 values; the range of c_2 is 1.120 (0.002) 1.140, 11 values. The five allowed values of a are 0.3 (0.1)

0.7. From the direct calculation, i.e., when the true structure of the rod is given, we know the conditions at the right end $x = 0.3$ which are $u(0.3) = 0.17028385$, $v(0.3) = 1.1266986$. The inverse calculations do not produce clearly the correct results $a_1 = a_2 = a_3 = 0.5$. It is believed that the grids of values of c_1 and of c_2 are not sufficiently fine, and that substantially improved results cannot be obtained without a great increase in computing expense. The computing time for the IBM 7044 is 1-1/2 minutes for these three stages. The one-dimensional reduction appears attractive in view of these results.

<u>One-dimensional dynamic programming approximation for the determination of the absorption coefficient.</u> The rod of unit length is divided into five sections of equal length 0.2. Armed with the internal measurements $w_i \approx u(x_i)$, we wish to determine the absorption coefficient of each slab. The correct choices are $a_N = 0.5$ for $N = 1, 2, \ldots, 5$. In the one-dimensional case, the only state variable is $c = v(b_N)$.

The outline immediately following lists the calculations for producing a_1, $f_1(c)$ and $r_1(c)$ for $N = 1$, and the next outline shown the general scheme, $N = 2, 3, \ldots,$.

ONE-DIMENSIONAL DYNAMIC PROGRAMMING

For stage $N = 1$

1. Print N
2. For each $c = v(b_N)$
    1. For each a
        1. Solve 2 point boundary-value problem for $v(0) = c'$.

           $$\begin{cases} \dot{u} = a\,v, & u(0) = 0 \\ -\dot{v} = a\,u, & v(0.2) = c \end{cases}$$

        2. Integrate to produce $u(x)$,

           $$\begin{cases} \dot{u} = a\,v, & u(0) = 0 \\ -\dot{v} = a\,u. & v(0) = c' \end{cases}$$

           and simultaneously calculate $d = \sum_i [u(x_i - w_i]^2$,

           and keep a running estimate of $f_i(c) \simeq \operatorname{Min}_a \{d\}$.
    2. Integrate to produce $r_1(c) = \rho(0.2)$,

       $\dot{\rho} = a(1 + \rho^2)$, $\rho(0) = 0$
    3. Print c, a_1, c', $r_1(c)$, $f_1(c)$

ONE DIMENSIONAL DYNAMIC PROGRAMMING

For each stage $N = 2, 3, 4, \ldots$

1. Print N
2. For each c
    1. For each a
        1. Produce $R = \rho(0.2)$, $T = t(0.2)$ by integration

           $\dot{\rho} = a(1 + \rho^2)$, $\rho(0) = 0$

           $\dot{t} = a \rho t$, $t(0) = 1$
        2. Solve the nonlinear equation for $c' = v(b_{N-1})$ and $r_{N+1}(c')$,

           $R c' r_{N-1}(c') = c' - c T$
        3. Solve 2 point boundary-value problem for $e' = u(0) = u(b_{N-1})$

           $\dot{u} = a v$, $v(0) = c'$

           $-\dot{v} = a u$, $v(0.2) = c$
        4. Integrate to produce $u(x)$

           $\dot{u} = a v$, $u(0) = e'$

           $-\dot{v} = a u$, $v(0) = c'$

           and calculate $d = \sum_i [u(x_i) = w_i]^2$
        5. Find $f_{N-1}(c')$ by interpolation
        6. Set $S(a) = d + f_{N-1}(c')$ and keep a running estimate of $f_N(c) \simeq \min_a \{S(a)\}$
    2. Integrate to produce $r_N(c) = \rho(0.2)$

       $\dot{\rho} = a(1 + \rho^2)$, $\rho(0) = r_{N-1}(c')$
    3. Print c, a_N, c', $r_N(c)$, $f_N(c)$
3. For each c
    1. Shift $f_N(c) \to f_{N-1}(c)$

To solve the nonlinear equation for $c' = v(b_{N-1})$ where $r(c')$ is known only on a grid of points, we compute the expressions $g_1 = R\,c'\,r_{N-1}(c')$, $g_2 = c' - c\,T$, and we take their difference $D = g_1 - g_2$. If $D = 0$, then c' has been found. Otherwise we repeat the procedure for each discrete value c_i', until the sign of D_i is opposite to that of D_{i-1}. We then interpolate linearly to find the quantity c' which makes $D = 0$. If the sign of D does not change, i.e., the curves g_1 and g_2 do not intersect, then the corresponding value of a is definitely not allowed to be the coefficient for the segment in question.

If the minimum cost $f_N(c)$ is large for a given state and all remaining states may be deleted from further consideration. This provides a saving in computing time, for each state to be considered requires many calculations. Of course, the precaution must be taken to order the c's properly, so that no potentially vital state is lost.

The proposed one-dimensional scheme has been tested numerically. The range of N is 1 to 3, the interfaces of the sections being located at $x = 0.2$, 0.4, and 0.6. The states $v(b_N) = c$ are 1.04955 (0.00015) 1.13385, 563 in all. This number is reduced in stage 2 to 546, by the use of the above test. Four values of the absorption coefficient a are allowed: 0.1, 0.3, 0.5, and 0.7. There are nine perfectly accurate observations of u per segment, a total of 27 data points. The integration method is Adams-Moulton with a grid size of 0.01.

From the output of our computation, we see that the minimum value of $f_3(c)$ is 0.387×10^{-8} and occurs when the input flux is $v(0.6) = c = 1.08855$ and the absorption coefficient for the segment of the medium between $x = 0.4$ and $x = 0.6$ is taken to be $a = 0.5$. This is very close to the true answer, $v(0.6) = 1.08860$, and the value of the parameter a is correct. The calculation tells us that the next state at $x = 0.4$ will be $v = 1.11673$. The nearest grid point in c is 1.11675, and the cost $f_2(1.11675)$ is indeed a minimum, 0.824×10^{-9}. The absorption coefficient for segment 2 is 0.5, the correct solution. The next state at $x = 0.2$ is predicted to be 1.13377. The nearest discrete state is 1.13385, possessing a cost $f_1(1.13385)$ $= 0.181 \times 10^{-9}$. The absorption coefficient is 0.5, again the correct answer. The solutions at each state are clearly found, the minimum cost differing from the others by at least an order of magnitude. These dynamic programming calculations of about 20 minutes have very accurately determined the input, and they have identified the medium.

Now we wish to test the one-dimensional method of determining the structure of the medium when the measurements are few and of limited accuracy. We consider the rod of length 0.8 consisting of 4 segments of equal length 0.2. There are again the same 563 discrete states in c, and the same four possible absorption coefficients 0.1, 0.3, 0.5 and 0.7. However, there are only three observations per segment and these are correct to only two significant figures. Knowing the inputs to the first three stages N = 1, 2, 3, we see from the output of the calculations that the absorption coefficients are $a_1 = a_2 = a_3 = 0.5$, the correct solution in this region. On the other hand, we are not able to accurately identify the input to a given segment on the basis of these calculations, because the minimum of the function f is broad and it is not centered at the correct value of the input c. For stage $N = 4$, the value of a_4 is determined to be 0.3, and incorrect value. These experiments might serve as a warning to the experimental investigator. They show that the processing of data with a small number of measurements requires higher accuracy than two figures, and that if the measurements are of limited accuracy, many measurements should be made. This trial consumes 34 minutes of IBM 7044 computing time. This time of calculation could be reduced greatly by streamlining the calculations. No attempt to do this was made here; feasibility was our sole concern.

For other approaches to transport theory, see Refs. 10-14.

REFERENCES

1. Chandrasekhar, S., Radiative Transfer, Dover Publications, Inc., New York, 1960.

2. Agranovich, Z. S., and V. A. Marchenko, The Inverse Problem of Scattering Theory, Gordon and Breach, New York, 1963.

3. Bellman, R. E., H. Kagiwada, R. E. Kalaba, and Sueo Ueno," On the Identification of Systems and the Unscrambling of Data - II: An Inverse Problem in Radiative Transfer," Proc. Nat. Acad. Sci., Vol. 53, pp. 910-913, May 1965.

4. Sims, A. R., "Ceratin Aspects of the Inverse Scattering Problem," J. Soc. Indust. Appl. Math., Vol. 5, pp. 183-205, 1957.

5. Bellman, R. E., Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

6. Bellman, R. E., and R. E. Kalaba, "Transport Theory and Invariant Imbedding," Nuclear Reactor Theory, pp. 206-2.8, edited by G. Birkhoff and E. Wigner, Amer. Math. Soc., Providence, Rhode Island, 1961.

7. Bellman, R. E., and R. E. Kalaba, "Functional Equations, Wave Propagation and Invariant Imbedding," J. Of Math. and Mech., Vol. 8, September 1959, pp. 683-704.

8. Bellman, R. E., and S. E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

9. Bellman, R. E., H. H. Kagiwada, and R. E. Kalaba, Numerical Studies of a Two-point Nonlinear Boundary Value Problem Using Dynamic Programming, Invariant Imbedding, and Quasilinearization, The RAND Corporation RM-4069-PR, March 1964.

10. Bellman, R., R. Kalaba, and G. M. Wing, "Invariant Imbedding and Variational Principles in Transport Theory," Bull. Am. Math. Soc., Vol. 67, No. 4, pp. 396-399, 1961.

11. Ash, M., R. Bellman, and R. Kalaba, "On Control of Reactor Shut-down Involving Minimal Xenon Poisoning," Nuclear Science and Engineering, Vol. 6, No. 2, pp. 152-156, 1959.

12. Harris, T. E., The Theory of Branching Processes, Springer-Verlag, Berlin, 1963.

13. Fujita, H., K. Kobayashi, and T. Hyodo, "Backscattering of Gamma Rays from Iron Slabs," Nuclear Science and Engineering, Vol. 19, pp. 437-440, 1964.

14. Shimizu, A., Application of Invariant Imbedding to Penetration Problems of Neutrons and Gamma Rays, NAIG Nuclear Research Laboratory, Research Memo-1, December 1963.

CHAPTER SIX

INVERSE PROBLEMS IN WAVE PROPAGATION: MEASUREMENTS OF TRANSIENTS

1. INTRODUCTION

The wave equation

(1) $$\Delta u = \frac{1}{c^2} u_{tt} ,$$

is one of the basic equations of mathematical physics. If we suppose that the local speed of propagation is a function of position

(2) $$c = c(x, y, z)$$

then the difficulties in studying the various initial and boundary value problems which arise are well known [1, 2, 3]. In the sections which follow, we wish to study some of the inverse problems which arise when we attempt to determine the properties of a medium on the basis of observations of a wave passing through the medium. Such problems are of central importance in such varied areas as ionospheric and tropospheric physics, seismology, and electronics. Some early results are due to Ambarzumian [4] and Borg [5].

We shall discuss some one-dimensional problems. Our basic technique is to reduce the partial differential equation in (1) to a system of ordinary differential equations either by using Laplace transforms or by considering the steady-state situation. Then our previously developed methodology is applicable For simplicity and specificity we shall employ the nomenclature associated with the problem of the vibrating string. In passing, we note that our methodology is applicable to the diffusion equation, to the telegrapher's equation, and to other similar propagation equations.

2. THE WAVE EQUATION

Consider an inhomogeneous medium which extends from $x = 0$ to $x = 1$, for which the wave equation

$$u_{tt} = c^2 u_{xx} \tag{1}$$

is applicable. In this equation, the disturbance $u(x,t)$ is a function of position and time. Let us assume that the wave speed c satisfies the equation

$$c^2 = a + bx , \tag{2}$$

where a and b are constants, as yet unknown, which are to be determined on the basis of experiments.

Let the initial conditions be

(3) $u(x, 0) = g(x)$,

(4) $u_t(x, 0) = v(x)$.

Let the boundary conditions be

(5) $u(0, t) = 0$,

(6) $Tu_x(1, t) = f(t)$.

Eqs. (1) – (6) may, for example, describe an inhomogeneous string, which is fixed at the end $x = 0$, while a force $f(t)$ is applied perpendicular to the string at $x = 1$, and T is the known tension.

The disturbance at the end $x = 1$, $u(1, t_i)$, is measured at n instants of time. On the basis of these observations, we wish to estimate the values of the parameters a and b, and thus to deduce the inhomogeneity of the medium.

3. LAPLACE TRANSFORMS

In order to reduce the partial differential wave equation to a system of ordinary differential equations, we take Laplace transforms of both sides of (2.1). We denote transforms by capital letters, for example,

(1) $U_s(x) = U(x,s) = L\{u(x,t)\}$.

Equation (1) becomes

(2) $s^2U(x,s) - su(x,0) - u_t(x,0) = c^2U_{xx}$.

Using (2.2) – (2.4), we obtain the desired system of ordinary differential equations,

(3) $$(a + bx)U_{xx} = s^2U(x,s) - sg(x) - v(x) ,$$

in which s is a parameter, $s = 1, 2., \ldots, N$. The boundary conditions are

(4) $$U(0,s) = 0, \quad TU_x(1,s) = F(s) .$$

The unknown constants a and b are to be determined by minimizing the expression

(5) $$\sum_{s=1}^{N} [U_{0bs}(1,s) - U(1,s)]^2 .$$

The quantities $U_{0bs}(1,s)$ are the Laplace transforms of the experimentally observed values $u(1,t_i)$, while the quantities $U(1,s)$ are the solutions of equations (3) and (4). The use of Gaussian quadrature [6] leads to the approximate formula for the Laplace transform of the observations,

(6) $$U_{0bs}(1,s) \cong \sum_{i=1}^{N} r_i^{s-1} u(1,t_i)w_i, \quad s = 1, 2, \ldots, N .$$

Similarly, the transform of the force may be produced with the use of the formula

(7) $$F(s) \cong \sum_{i=1}^{N} r_i^{s-1} f(t_i)w_i, \quad s = 1, 2, \ldots, N .$$

In these equations, r_i are the roots of the shifted Legendre polynomial $P_N^*(x) = P_N(1 - 2x)$ and w_i are the related weights. In addition, the times of evaluation are

(8) $t_i = -\log_e r_i$, $i = 1, 2, \ldots, N$.

Interpolation may be necessary in order to have the data for these special times. After the solution has been found for $U(x,s)$, the inverse transforms $u(x,t)$ may be obtained by a numerical inversion method [7].

4. FORMULATION

The constants a and b are to be thought of as functions of x which satisfy the differential equations $a_x = 0$, $b_x = 0$. The complete system of equations for this nonlinear boundary value problem is

$$U_{xx} = \frac{1}{a+bx}\,[s^2U(x,s) - sg(x) - v(x)] \ , \quad s=1, 2, \ldots, N \ ,$$

$$a_x = 0 \ ,$$

$$b_x = 0 \ .$$

This is equivalent to a system of $2N + 2$ first order equations, so there must be $2N + 2$ boundary conditions. These conditions are

(2) $U(0,s) = 0$, $s = 1, 2, \ldots, N$,

(3) $U_x(i,s) = \dfrac{F(s)}{T}$, $s = 1, 2, \ldots, N$,

(4) $$\frac{\partial}{\partial a}\left\{\sum_{s=1}^{N}\left[U_{Obs}(1,s) - U(1,s)\right]^2\right\} = 0 ,$$

(5) $$\frac{\partial}{\partial b}\left\{\sum_{s=1}^{N}\left[U_{Obs}(1,s) - U(1,s)\right]^2\right\} = 0 .$$

5. SOLUTION VIA QUASILINEARIZATION

The nonlinear boundary value problem may be resolved using the technique of quasilinearization [8, 9, 10]. In each step of the successive approximation method, we must solve the linear differential equations

(1) $$\begin{aligned} \frac{dU_s^n}{dx} &= W_s^n , \\ \frac{dW_s^n}{dx} &= \frac{s^2 U_s^n}{a+bx} - \frac{a^n+b^n x}{(a+bx)^2} s^2 U_s + \frac{s^2 U_s}{a+bx} , \\ \frac{da^n}{dx} &= 0 \\ \frac{db^n}{dx} &= 0,, \end{aligned}$$

where the superscripts n indicate the solution in the n^{th} approximation, while the un-superscripted variables belong to the $(n-1)^{st}$ approximation. The boundary conditions are

(2) $$U_s^n(0) = 0 ,$$

(3) $$W_s^n(1) = \frac{F(s)}{T} ,$$

(4) $$\frac{\partial}{\partial a^n}\left\{\sum_{s=1}^{N} [U_{Obs}(1,s) - U_s^n(1)]^2\right\} = 0 \ .$$

(5) $$\frac{\partial}{\partial b^n}\left\{\sum_{s=1}^{N} [U_{Obs}(1,s) - U_s^n(1)]^2\right\} = 0 \ .$$

We represent the solution in the n^{th} approximation as a linear combination of a particular vector solution and $N + 2$ homogeneous vector solutions. If we let the column vector $X(x)$ represent the solution in the n^{th} approximation, where the components of X are $(U_1^n, U_2^n, \ldots, U_N^n, W_1^n, W_2^n, \ldots, W_N^n, a^n, b^n)$, and if we let the column vectors $P(x)$, $H^1(x)$, $H^2(x), \ldots, H^{N+2}(x)$ represent the particular and homogeneous solutions, then we may write

(6) $$X(x) = P(x) + \sum_{i=1}^{N+2} H^i(x)y_i \ .$$

Since the system of differential equations is of order $2N + 2$, and since N initial conditions are prescribed, there are $N + 2$ missing initial conditions, represented by the $N + 2$ dimensional column vector Y,

(7) $$Y = (W_1^n(0), W_2^n(0), \ldots, W_N^n(0), a^n(0), b^n(0))^T \ .$$

The particul r and homogeneous solutions are computationally produced. In terms of these, the boundary conditions (2) – (5) require the solution of system of $N + 2$ linear algebraic equations,

(8) $$A\,Y = B\,,$$

where the elements of matrix A are

(9)
$$\begin{aligned}
A_{ij} &= H^{j}_{N+1}(1)\,, & i &= 1, 2, \ldots, N\,, \\
&= \sum_{s=1}^{N} H^{j}_{s}(1)\, H^{i}_{s}(1)\,, & i &= N+1, N+2\,,
\end{aligned}$$

$$j = 1, 2, \ldots, N, N+1, N+2,$$

and where the components of vector B are

$$\begin{aligned}
B_i &= \frac{F(i)}{T} - P_{N+i}(1)\,, & i &= 1, 2, \ldots, N\,, \\
&= \sum_{s=1}^{N} [U_{Obs}(1,s) - P_s(1)]\, H^{i}_{s}(1), & i &= N+1, N+2\,.
\end{aligned}$$

The method is applied iteratively for a fixed number of stages, about five, or it may be terminated when the approximations converge or diverge. The displacement function $u(x,t)$ may be obtained from its transform by a numerical inversion method of Bellman, et al. [7].

6. EXAMPLE 1 - HOMOGENEOUS MEDIUM, STEP FUNCTION FORCE

In this and the following example, we consider a homogeneous medium and make use of the analytical solution. In Example 3, we consider the more general problem of an inhomogeneous medium characterized by two unknown constants.

Consider the case in which we have a constant speed c which is given by the equation

(1) $$c^2 = a = 1 .$$

The value of T is unity, the input $f(t)$ is the Heaviside unit step function, $H(t)$, and the initial conditions are $g(x) = v(x) = 0$. The wave equation for the function $U(x,s)$ is

(2) $$U_{xx} = \frac{s^2}{c^2} U(x,s) .$$

The solution which satisfies (2), as well as the boundary conditions $U(0,s) = 0$, $TU_x(1,s) = F(s)$ is

(3) $$U(x,s) = \frac{c\ F(s)\ \sinh \frac{s}{c} x}{T\ s\ \cosh \frac{s}{c}} .$$

Noting that the Laplace transform of the force is

(4) $$F(s) = L\{H(t)\} = \frac{1}{s} ,$$

we may explicitly evaluate U at the boundary $x = 1$, and we obtain the values

(5) $$U(1,s) = \frac{c}{T} \frac{1}{s^2} \tanh \frac{s}{c} = \frac{1}{s^2} \tanh s .$$

The inverse transform,

(6) $$u(1,t) = L^{-1}\left\{\frac{c}{T} \frac{1}{s^2} \tanh \frac{s}{c}\right\} ,$$

is shown in Fig. 1.

We decide to use a seven point quadrature, so that $N = 7$. Making use of the known solution, we "produce" the observations at the specified times t_i, which are listed in Table 1.

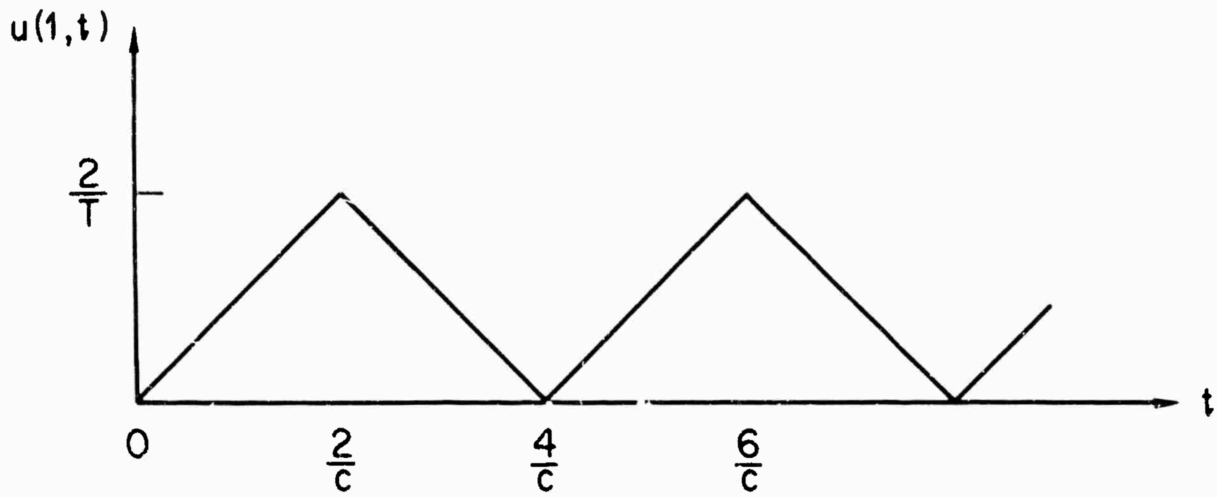


Fig. 1. The analytical solution of the wave equation at $x = 1$, with a step function input: $u(1,t) = L^{-1}\left\{\frac{c}{T}\frac{1}{s^2}\tanh\frac{s}{c}\right\}$.

TABLE 1

SEVEN OBSERVATIONS FOR EXAMPLE 1

t_i	$u(1,t_i)$
3.671195	0.328805
2.046127	1.953873
1.213762	1.213762
0.693147	0.693147
0.352509	0.352509
0.138382	0.138382
0.025775	0.025775

The approximate transforms U_{0bs} are computed using the formula from Gaussian quadrature. In Table 2, these quantities are compared with the exact transforms using the analytical solution. The transforms of the input, $F(s)$, are computed with the aid of the approximate formula. The approximate transforms, $U_{0bs}(1,s)$ and $F(s)$, are used in the calculations because in the general case, the analytical transforms will be unobtainable.

TABLE 2

THE LAPLACE TRANSFORMS $U_{0bs}(1,s)$ FOR EXAMPLE 1

s	Approximate $U_{0bs}(1,s)$	Exact $U_{0bs}(1,s)$
1	0.759442	0.761594
2	0.242907	0.241007
3	0.110753	0.110561
4	0.0624686	0.064580
5	0.0399963	0.0399964
6	0.0277773	0.0277774
7	0.0204081	0.0204081

There are only $2N + 1$ variables in this example, so that when $N = 7$, we have a solution of dimension 15. During each stage of the calculations, we have to produce a particular solution and $N + 1 = 8$ homogeneous solutions, i.e., $15 \times 9 = 135$ differential equations must be integrated. For the initial conditions on P, we choose $P(0)$ identically zero. We also choose for $H^j(0)$, the unit vector which has all of its components zero except the $(N+j)^{th}$, which is unity. Any linear combination of these P and H vectors identically satisfies the conditions $U_s(0) = 0$, $s = 1, 2, \ldots, N$. For the remaining boundary conditions, we must invert the 8x´ matrix A.

As a first check case, we try an initial approximation $a^0 = 1$ which is the correct value of a. We estimate the initial slopes to be $W_s(0) = 10^{-3}$. The initial approximation is generated by integrating the nonlinear equations (31) with this set of estimates, as initial conditions. In three iterations we obtain better estimates of the slopes $W_s(0)$, but the value of a has drifted to 1.00023. This value may be used as a comparison for other trials. The results of three experiments are shown in the following table. The initial approximation a^0 is listed in Table 3, followed by the successive approximations a^n, $n = 1, 2, \ldots,$ for each of the three trials.

TABLE 3

SUCCESSIVE APPROXIMATIONS OF THE VELOCITY a IN EXAMPLE 1

Approximation	Run 1	Run 2	Run 3
0	1.2	1.5	0.5
1	1.00991	0.46752	0.50922
2	1.00186	0.48284	0.80612
3	1.00018	0.67047	0.97736
4	———	0.89366	1.00049
5	———	0.99110	1.00022
6	———	1.00041	———

In Run 2, U and U_x at $x = 1.0$ are consistent to two significant figures with the conditions. In Run 3 U is in agreement with the observations to four places, and U_x agrees with the conditions to five figures. Recall that the conditions on U_x are supposed to be exact, and those on the $U(1,s)$ are of a least squares nature, which may help to explain why U_x is in better agreement than U for Run 3.

7. EXAMPLE 2 – HOMOGENEOUS MEDIUM, DELTA-FUNCTION FORCE

In Example 2, we have a homogeneous medium and zero initial conditions. The boundary conditions are again $u(0,t) = 0$, $u_x(1,t) = f(t)$, where now the input is $f(t) = \delta(t)$, the delta function. The Laplace transform of the delta function is $F(s) = 1$. The analytical solution for $x = 1$ is

(1) $$u(1,t) = L^{-1}\left\{\frac{c}{T}\frac{1}{s}\tanh\frac{s}{c}\right\} .$$

This function is sketched in Fig. 2, for the case $c = 1$, $T = 1$.

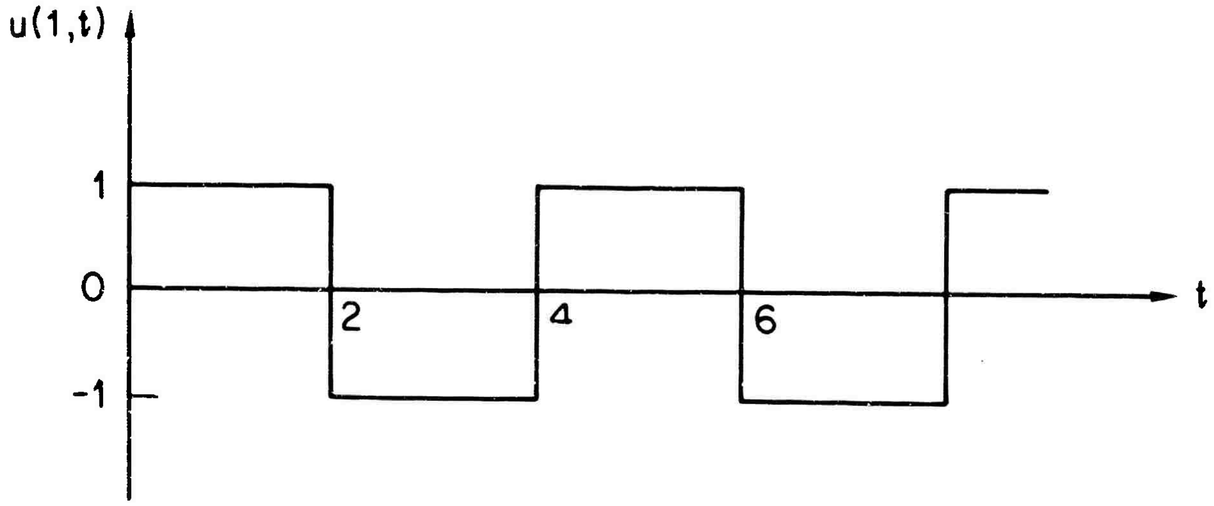


Fig. 2. The analytical solution of the wave equation at $x = 1$, with a delta function input: $u(1,t) = L^{-1}\left\{\frac{1}{s}\tanh s\right\}$.

We again take $N = 7$. The observations are

(1) $$u(1,t_i) = 1 , \quad \text{for } i = 1, 2, \ldots, 5 ,$$
$$= -1 , \quad \text{for } i = 6, 7 .$$

The transforms of the observations, $U_{Obs}(1,s)$, are computed using the quadrature approximation. A comparison of these values against the exact transforms using (1) is given in Table 4.

TABLE 4

THE LAPLACE TRANSFORMS $U_{0bs}(1,s)$ for EXAMPLE 2

s	Approximate $U_{0bs}(1,s)$	Exact $U_{0bs}(1,s)$
1	0.59080964	0.76159415
2	0.46055756	0.48201379
3	0.32857798	0.33168492
4	0.24939415	0.24983232
5	0.19992192	0.19998184
6	0.16665658	0.16666462
7	0.14285584	0.14285690

All initial approximations in the following experiments are produced by integrating the nonlinear equations with a [illegible]te set of estimated initial conditions. The check case with initial approximation $a^0 = 1$, a correct guess, results [illegible] a convergence to the wrong value $a \cong 0.9$. With $a^0 = 0.5$, the estimate is again 0.9. With $a^0 = 1.5$, the value -0.8 is obtained. It is suspected that the discontinuous nature of the function $u(1,t)$ is the cause of the difficulty in determining a. A more reasonable formulation of the problem should include damping terms to overcome this obstacle. In spite of the poor estimates of a in the first two trials, the final approximations are quite close to the exact observations $U_{0bs}(1,s)$, rather than the approximate, and the conditions $U_x(1,s) = F(s)/T$ are met, to within 0.001%.

8. EXAMPLE 3 - INHOMOGENEOUS MEDIUM WITH DELTA-FUNCTION INPUT

As an example of the inverse problem for an inhomogeneous medium as originally posed, consider the case in which the wave velocity is indeed given by the equation

$$c^2 = a + bx , \tag{1}$$

where $a = 1$ and $b = 0.5$. We again set the initial conditions $u(x,0) = u_t(x,0) = 0$, and the tension $T = 1$. We exert a delta-function force, $f(t) = \delta(t)$, on the boundary $x = 1$, and we observe the displacement $u(1,t)$ as a function of time. Laplace transforms $U_{0bs}(1,s)$ are computed. The parameters a and b are determined for best agreement with these transforms of observations.

In this study, the experimenter obtains his data with the use of the digital computer, rather than by the actual performance of laboratory experiments. The exact solution for this inhomogeneous wave problem is not readily available analytically. We must produce the solution computationally, by solving the wave equation with its boundary conditions. Since we prefer to deal with the ordinary differential equation for the function $U_s(x)$, we solve the approximately equivalent linear two-point boundary value problem

$$U_{xx} = \frac{1}{a+bx} s^2U(x,s) , \tag{2}$$

(3) $U(0,s) = 0$,

(4) $U_x(1,s) = 1$,

for $s = 1, 2, \ldots, N$. We produce a particular solution and N independent homogeneous solutions which, when combined to satisfy conditions (3) and (4), also produce the data of Table 5. These are the "observations".

TABLE 5

THE LAPLACE TRANSFORMS $U_{0bs}(1,s)$ FOR EXAMPLE 3

s	$U_{0bs}(1,s)$
1	.811967
2	.551174
3	.390695
4	.297835
5	.239837
6	.200613
7	.172392

These quantities $U_{0bs}(1,s)$ can be inverted numerically to produce the function $u(1,t_i) = L^{-1}\left\{U_{0bs}(1,s)\right\}$, which are the observations of the disturbance in the space of x and t. However, we need the set of transforms for use in determining the parameters a and b, and so we decide to utilize these numbers directly, as they appear in the table.

Two series of experiments are performed (see Tables 6 and 7). In one, the observations are given correct to 6 significant figures, and the initial approximations are varied. The true values of the unknown parameters are $a = 1.0$ and $b = 0.5$.

TABLE 6

SERIES I RESULTS FOR EXAMPLE 3

Observations are correct to six significant figures.

Run 1	$a^0 = .9$ $a^3 = .9998$	$b^0 = .6$ $b^3 = .5002$	$W_s^0(0)$ correct to 1 figure
Run 2	$a^0 = 1.2$ $a^3 = .9998$	$b^0 = .3$ $b^3 = .5002$	$W_s^0(0)$ correct to 1 figure
Run 3	$a^0 = 1.2$ $a^5 = .9996$	$b^0 = .3$ $b^5 = .50005$	$W_s^0(0) = .05$

TABLE 7

SERIES II RESULTS FOR EXAMPLE 3

Observations are in error by specified amounts

Run 4	$a^0 = 1.2$ $a^3 = .9872$	$b^0 = .3$ $b^3 = .5182$	$W_s^0(0)$ correct to 1 figure; Observations; $\pm 1\%$ error
Run 6	$a^0 = 1.2$ $a^3 = .937$	$b^0 = .3$ $b^3 = .590$	$W_s^0(0)$ correct to 1 figure; Observations: $\pm 5\%$ error

In the series I experiments, with accurate observations, rapid convergence to the correct values of the parameters occurs. The higher approximations of the initial slopes $W_s(0)$ are not listed, but these are considerably improved values.

In the series II experiments, noisy observations are used. For example in Run 4, the observations $U_{0bs}(1,s)$ are in error by the relative amounts +1%, −1%, +1%, ..., +1% for $s = 1, 2, 3, \ldots, 7$ respectively. The relative errors in the third approximations $a^3 = .9872, b^3 = .5182$, are 1.3% and 3.6% respectively. The results of this trial may be contrasted with the final approximations of Run 2. In Run 2, observations which are correct to six significant figures produce values of the parameters which are correct to less than 0.04%. Run 4 may also be compared with Run 6, in which case we are comparing the effect of 1% errors against 5% errors. The results of Run 6 involve errors of −6% in the value of a, and +18% in b.

The time required for these calculations is about one-half minute per iteration, with the IBM 7044. Each iteration includes the integration of $(N+3)(2N+2) = 10 \times 16 = 160$ differential equations, and the inversion of a 9x9 matrix. The FORTRAN programs for all of the cases treated are to be found in Appendix F.

9. DISCUSSION

The methods presented here are of practical use in identifying a system described by a wave equation or by linear differential equations or by a weighting function [11].

REFERENCES

1. Courant, R., and D. Hilbert, Methods of Mathematical Physics, Interscience Publishers, Inc., New York, Vol. I, 1953, Vol. II, 1962.

2. Frank, P. and R. von Mises, Die Differential- und Integralgleichungen, Mary S. Rosenberg, N. Y., 1943.

3. Brillouin, L., Wave Propagation in Periodic Structures, Dover Publications, Inc., 1953.

4. Ambarzumian, V., "Über Einige Fragen der Eigenwerttheorie," Zeitschrift für Physik, Vol. 53, pp. 690-695, 1929.

5. Borg, G., "Eine Umkehrung der Sturm-Liouville Eigenwertaufgabe Bestimmung der Differentialgleichung Durch die Eigenwerte," Acta Math., Vol. 78, pp. 1-96, 1946.

6. Bellman, R., R. Kalaba, and M. Prestrud, Invariant Imbedding and Radiative Transfer in Slabs of Finite Thickness, American Elsevier Publishing Co., Inc., New York, 1963.

7. Bellman, R., H. Kagiwada, R. Kalaba, and M. C. Prestrud, Invariant Imbedding and Time-Dependent Transport Processes, American Elsevier Publishing Co., Inc., New York, 1964.

8. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," J. Math. and Mech., Vol. 8, pp. 519-574, 1959.

9. Bellman, R. E., and R. E. Kalaba, Quasilinearization and Boundary Value Problems, American Elsevier Publishing Co., New York, to appear.

10. Bellman, R., H. Kagiwada, and R. Kalaba, "Orbit Determination as a Multipoint Boundary-value Problem and Quasilinearization," Proc. Nat. Acad. Sci. USA, Vol. 48, pp. 1327-1329, 1962.

11. Bellman, R., H. Kagiwada and R. Kalaba, ""Identification of Linear Systems via Numerical Inversion of Laplace Transforms", IEEE Transactions on Automatic Control, Vol. AC-10, No. 1, pp. 111-112, January, 1965.

BLANK PAGE

CHAPTER SEVEN

INVERSE PROBLEMS IN WAVE PROPAGATION: MEASUREMENTS OF STEADY STATES

1. INTRODUCTION

Consider the propagation of waves in a plane parallel stratified medium [1-5] extending from $x = 0$ to $x = b$, with index of refraction $n(x)$ varying continuously throughout the slab. The slab is bounded by a vacuum to the left $(n_0 = 1)$ and a homogeneous medium with index of refraction n_1 to the right, as shown in Fig. 1. We assume a lossless dielectric medium in which $n(x)$ is independent of frequency.

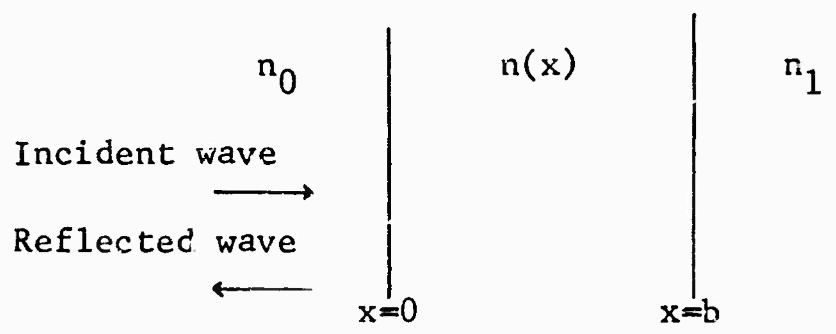


Fig. 1. The physical situation

The wave equation is

(1) $$k^2(x)u_{tt} = u_{xx} ,$$

where k is the wave number. The wave number is related to the index of refraction by the formula

(2) $$k(x) = \frac{\omega}{c_0} n(x) ,$$

where c_0 is the speed of light in a vacuum, which we normalize to unity, and ω is the angular frequency. We are interested in solutions of the form

(3) $$u(x,t) = e^{-i\omega t}u(x) ,$$

corresponding to the steady-state case where the transients have died down. The function $u(x)$ satisfies the ordinary differential equation.

(4) $$u''(x) + \omega^2 k^2(x)u(x) = 0 .$$

We shall often neglect the function $e^{-i\omega t}$ in all of the solutions, and speak of the functions $u(x)$ as waves.

We conduct a series of experiments in which waves of different frequencies ω_i are normally incident in the medium from the left, i.e., the incident wave is $e^{i(k_0x-\omega_i t)}$, or simply e^{ik_0x} . The reflected waves at each frequency are observed. We wish to determine the index of refraction $n(x)$ throughout the slab on the basis of these measurements.

2. SOME FUNDAMENTAL EQUATIONS [1, 2]

Consider the case of two adjacent homogeneous media, as illustrated in Fig. 2.

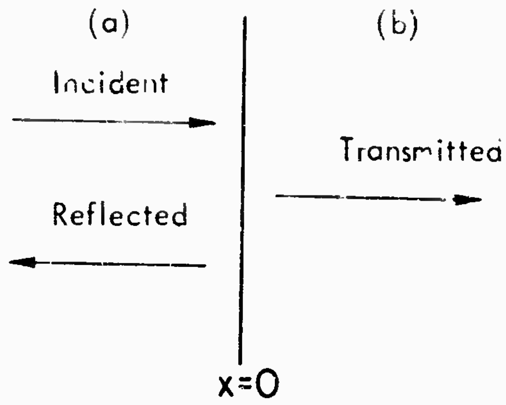


Fig. 2. Waves at an interface.

A plane wave of frequency ω traveling in medium (a) is incident at the interface $x = 0$. Let the wave numbers of medium (a) and medium (b) be k_a and k_b respectively. The incident wave is $e^{ik_a x} e^{-i\omega t}$ and the reflected wave is $re^{-ik_a x} e^{-i\omega t}$ where

$$r = \frac{k_a - k_b}{k_a + k_b} . \tag{1}$$

The transmitted wave is $te^{ik_b x} e^{-i\omega t}$, where

$$t = \frac{2k_a}{k_a + k_b} . \tag{2}$$

Now consider the case of two interfaces between three homogeneous media, (a), (b), and (c). The interfaces are separated by a distance Δ.

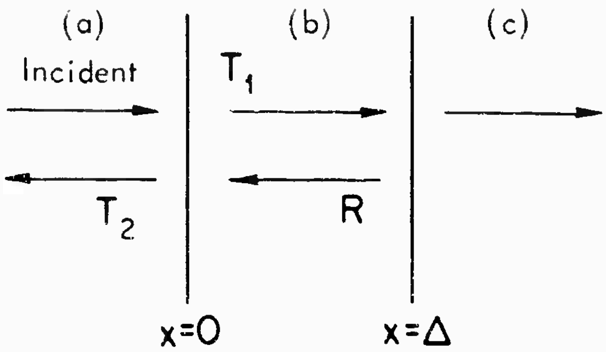


Fig. 3. Waves at two interfaces.

The incident wave is again $e^{ik_a x} e^{-i\omega t}$. The wave which is transmitted through $x = 0$, reflected at $x = \Delta$, and transmitted again through $x = 0$ is $ve^{-ik_a x} e^{-i\omega t}$, where

$$(3) \qquad v = \frac{2k_a}{k_a + k_b} \cdot \frac{k_b - k_c}{k_b + k_c} e^{2ik_b\Delta} \cdot \frac{2k_b}{k_b + k_a} + o(\Delta) ,$$

and $o(\Delta)$ includes the terms proportional to the second and higher powers of Δ. This equation shows how v depends on frequency by means of the exponential factor $e^{2ik_b\Delta} = e^{2i\omega n_b\Delta}$.

3. INVARIANT IMBEDDING AND THE REFLECTION COEFFICIENT [5]

Now we turn our attention to the reflection coefficient r as a function of thickness of the medium. We assume that the slab is inhomogeneous and that it extends from $x = z$ to $x = b$. The right boundary $x = b$ is to be considered fixed, while the left boundary $x = z$ is variable, as shown in Fig. 4. The incident wave is $e^{ik(z\_)(x-z)}$, deleting the time dependent factor $e^{-i\omega t}$, where $k(z\_) = k(z-0)$ is the wave number of the homogeneous medium to the left, and where the expression $e^{ik(z\_)(x-z)}$ is used rather than $e^{ik(z\_)x}$ in order to normalize the incoming intensity at $x = z$.

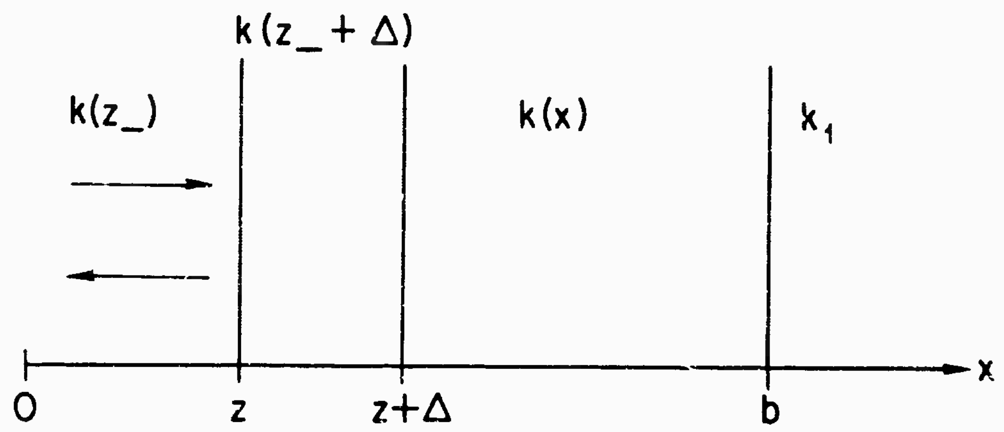


Fig. 4. An inhomogeneous medium of thickness $b-z$.

The reflected wave is $r(z)e^{-ik(z\_)(x-z)}$.

Using the technique of invariant imbedding, we relate the reflection coefficient for a slab extending from z to b to that for a slab extending from $z + \Delta$ to b. The reflected wave may be expressed, to terms of order zero and one in Δ, as arising from three processes:

(a) immediate reflection at z ;

(b) transmission through the interface at $x = z$, reflection at $z + \Delta$ from the slab $(z + \Delta, b)$, and transmission through z ;

(c) transmission through the interface at $z = z$, reflection at $z + \Delta$ from the slab $(z + \Delta, b)$, reflection at z, reflection at $z + \Delta$, and finally transmission through z.

These three cases are represented in Fig. 5.

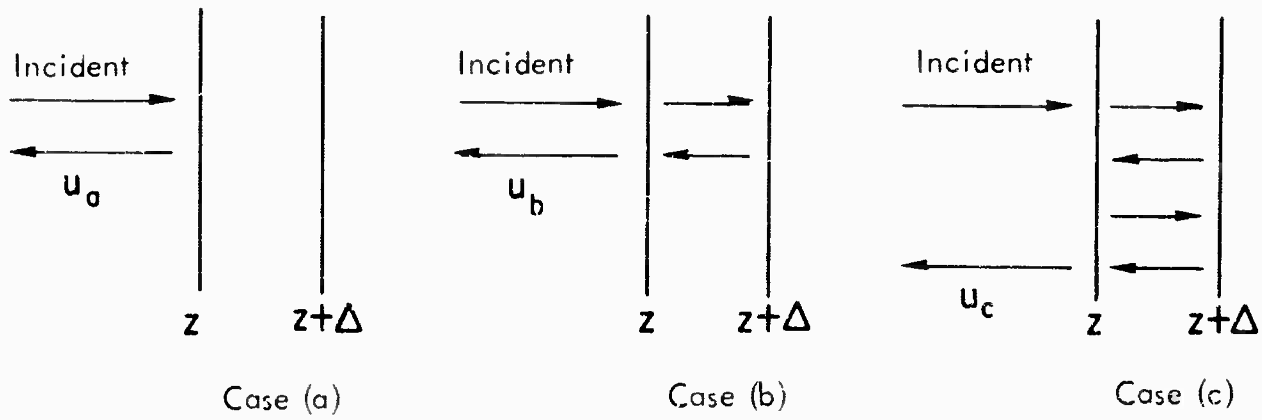


Fig. 5. Three processes in a stratified slab.

The wave which is reflected from the slab (z,b) is

(1) $$r(z)e^{-ik(z_-)(x-z)} = [r_a+r_b+r_c+o(\Delta)]e^{-ik(z_-)(x-z)} ,$$

where

(2) $$r_a = \frac{k(z_-) - k(z_-+\Delta)}{k(z_-) + k(z_-+\Delta)},$$

(3) $$r_b = \frac{2k(z_-)}{k(z_-) + k(z_-+\Delta)} \cdot r(z+\Delta)e^{2ik(z_-+\Delta)\Delta} \cdot \frac{2k(z_-+\Delta)}{k(z_-+\Delta) + k(z_-)} ,$$

$$r_c = \frac{2k(z_-)}{k(z_-) + k(z_- + \Delta)} \cdot r(z+\Delta)e^{2ik(z_- + \Delta)\Delta}$$

(4)
$$\cdot \frac{k(z_- + \Delta) - k(z_-)}{k(z_- + \Delta) + k(z_-)} \cdot r(z+\Delta)e^{2ik(z_- + \Delta)\Delta}$$

$$\cdot \frac{2k(z_- + \Delta)}{k(z_- + \Delta) + k(z_-)} ,$$

and $k(z_- + \Delta) = k(z+\Delta-0)$ is the wave number in the region immediately to the left of the interface $z+\Delta$. Simplifying to terms of order Δ, we have

$$r(z) = \frac{k(z_-)-k(z_- + \Delta)}{[k(z_-)+k(z_- + \Delta)]} + \frac{4k(z_-)k(z_- + \Delta)}{[k(z_-)+k(z_- + \Delta)]^2} [1+2ik(z_- + \Delta)\Delta] r(z+\Delta)$$

(5)
$$- \frac{4k(z_-)k(z_- + \Delta)}{[k(z_-)+k(z_- + \Delta)]^2} \cdot \frac{[k(z_-)-k(z_- + \Delta)]}{[k(z_-)+k(z_- + \Delta)]} [1+4ik(z_- + \Delta)\Delta] r^2(z+\Delta)$$

$$+ o(\Delta) .$$

Making use of the formula for the derivative of r,

(6)
$$\frac{dr}{dz} = \lim_{\Delta \to 0} \frac{r(z+\Delta)-r(z)}{\Delta} ,$$

we obtain the Riccati equation

(7)
$$\frac{dr}{dz} = \frac{k'}{2k} - 2ikr - \frac{k'}{2k} r^2 .$$

The "initial" condition reduces to the formula for an interface between two media

(8)
$$r(b) = \frac{k(b-0)-k_1}{k(b-0)+k_1} .$$

In terms of the index of refraction, Eqs. (7) and (8) are

(9) $$\frac{dr}{dz} = \frac{n'}{2n} - 2in\omega r - \frac{n'}{2n} r^2 ,$$

where $n = n(z)$, and

(10) $$r(b) = \frac{n(b)-n_1}{n(b)+n_1} .$$

The reflection coefficient for any inhomogeneous slab in which n varies as a function of x may be found by a simple (numerical) integration of (9) with the given initial condition (10). The integration is carried out from the right boundary $z = b$ to the left boundary $z = 0$.

4. PRODUCTION OF OBSERVATIONS

In place of performing laboratory experiments for obtaining reflection data [6, 7], we produce the observations computationally, for N different frequencies. The incident waves are $e^{i\omega_j n_0 x}$, and the reflected waves are $r_j(0)e^{-i\omega_j n_0 x}$. $j = 1, 2, \ldots . N$. We solve the initial value problems

(1) $$\frac{dr_j}{dz} = \frac{n'}{2n} - 2in\omega_j r_j - \frac{n'}{2n} r_j^2 ,$$

(2) $$r_j(b) = \frac{n(b)-n_1}{n(b)+n_1} , \; b \geq z \geq 0 ,$$

for the desired coefficients $r_j(0)$.

Since r_j is a complex reflection coefficient, we let

(3) $$r_j = R_j + iS_j ,$$

where R_j and S_j are real functions which satisfy the equations

(4) $$\frac{dR_j}{dz} = \frac{n'}{2n} + 2n\omega_j S_j - \frac{n'}{2n}(R_j^2 - S_j^2) ,$$
$$\frac{dS_j}{dz} = -2n\omega_j R_j - \frac{n'}{n} R_j S_j ,$$

(5) $$R_j(b) = \frac{n(b)-n_1}{n(b)+n_1} , \quad S_j(b) = 0 ,$$

for $j = 1, 2, \ldots, N$.

For the numerical experiment, we take

(6) $$n(x) = a_1 + a_2(x-1)^2$$

where $a_1 = 1$, $a_2 = 0.5$. We also choose

(7)
$$b = 1 ,$$
$$N = 3 ,$$
$$\omega_1 = 2\pi ,$$
$$\omega_2 = 4\pi ,$$
$$\omega_3 = 6\pi .$$

We assume that $n_1 = n(b)$, so that $R_j(b) = 0$.

We have chosen to normalize the speed,

(8) $$c_0 = 3 \times 10^{10}\text{cm/sec} = \frac{\text{one length unit}}{\text{one time unit}} .$$

We have chosen

$$(9) \qquad b = 1 \text{ length unit}$$

and we set

$$(10) \qquad \begin{aligned} b &= 3 \text{ cm} \\ &\cong 1 \text{ X-band microwave wave length}. \end{aligned}$$

Then

$$(11) \qquad 1 \text{ length unit} = 3 \text{ cm}$$

and

$$(12) \qquad 1 \text{ time unit} = 10^{-10} \text{ sec.}$$

To produce R_j and S_j, the real and imaginary parts of the reflection coefficients, for incident waves of frequencies 10, 20, and 30 kilo megacycles, we integrate Eqs. (4) with initial conditions $R_j(1) = 0$, $S_j(1) = 0$, for $j = 1, 2, 3$. We use a step length of $-.00$ and the Adams-Moulton integration scheme. The values $R_j(0)$, and $S_j(0)$ are the "observed" reflection coefficients. These are

$$(13) \qquad \begin{aligned} R_1(0) &= .13217783 \times 10^{-2}, & S_1(0) &= .14843017 \times 10^{-1}, \\ R_2(0) &= .32313148 \times 10^{-3}, & S_2(0) &= .95414704 \times 10^{-2}, \\ R_3(0) &= .38854984 \times 10^{-3}, & S_3(0) &= .58976205 \times 10^{-2}. \end{aligned}$$

5. DETERMINATION OF REFRACTIVE INDEX

We consider the inhomogeneous slab extending from $x = 0$ to $x = 1$. We are given observations of the real and imaginary parts of the reflection coefficients, $A_i \cong R_i$, $B_i \cong S_i$, where

$$(1) \qquad \begin{aligned} A_1 &= .132178 \times 10^{-2}, & B_1 &= .148430 \times 10^{-1}, \\ A_2 &= .323131 \times 10^{-3}, & B_2 &= .954147 \times 10^{-2}, \\ A_3 &= -.388550 \times 10^{-3}, & B_3 &= .589762 \times 10^{-2}, \end{aligned}$$

which correspond to frequencies $\omega_1 = 10$, $\omega_2 = 20$, and $\omega_3 = 30$ kilomegacycles [6, 7]. We seek to determine the values of the constants a and b in the equation for the index of refraction as a function of position,

$$(2) \qquad n(x) = a + b(x-1)^2,$$

in such a manner as to minimize the expression

$$(3) \qquad S = \sum_{i=1}^{3} [(A_i - R_i(0))^2 + (B_i - S_i(0))^2] .$$

The form S is the sum of squares of deviations between the solution of Eqs. (4.4) and (4.5), and the (perhaps inaccurate) observations (1).

The system of nonlinear equations is

$$(4)\ldots \qquad R_j' = \frac{n'}{2n} + 2n\omega_j S_j - \frac{n'}{2n}(R_j^2 - S_j^2)$$

$$S_j' = -2n\omega_j R_j - \frac{n'}{n} R_j S_j, \quad j = 1, 2, 3,$$

(4) $$a' = 0 \ ,$$

$$b' = 0 \ ,$$

where

(5) $$n = a + b(x-1)^2 \ ,$$

and

(6) $$n' = 2b(x-1) \ .$$

We obtain a system of linear differential equations by applying quasilinearization [8]. In the following linear equations, so as not to clutter the equations with superscripts indicating the approximations and subscripts indicating the components, we write the variables of the current k^{th} approximation as R, S, a, b, (also n and n'). Corresponding quantities in the previous $(k-1)^{st}$ approximation are ρ, σ, α, β (and η and η'). The linear equations obtained via quasilinearization are

(7)
$$\begin{aligned} R' &= \frac{\eta'}{2\eta} + 2\eta\omega\sigma - \frac{\eta'}{2\eta}(\rho^2 - \sigma^2) \\ &+ (R-\rho)\left(-\frac{\eta'}{\eta}\rho\right) + (S-\sigma)\left(2\eta\omega + \frac{\eta'}{\eta}\sigma\right) \\ &+ (a-\alpha)\left[\frac{1}{2}\frac{\partial}{\alpha\alpha}\left(\frac{\eta'}{\eta}\right) + 2\omega\sigma\frac{\partial\eta}{\partial\alpha} - \frac{1}{2}(\rho^2-\sigma^2)\frac{\partial}{\partial\alpha}\left(\frac{\eta'}{\eta}\right)\right] \\ &+ (b-\beta)\left[\frac{1}{2}\frac{\partial}{\partial\beta}\left(\frac{\eta'}{\eta}\right) + 2\omega\sigma\frac{\partial\eta}{\partial\beta} - \frac{1}{2}(\rho^2-\sigma^2)\frac{\partial}{\partial\beta}\left(\frac{\eta'}{\eta}\right)\right] \ , \end{aligned}$$

$$
\begin{aligned}
S' &= -\,2\eta\omega\rho - \frac{\eta'}{\eta}\rho\sigma \\
&+ (R-\rho)(-2\eta\omega - \frac{\eta'}{\eta}\sigma) + (S-\sigma)(-\frac{\eta'}{\eta}\rho) \\
(8) \qquad &+ (a-\alpha)[-2\omega\rho\frac{\partial\eta}{\partial\alpha} - \rho\sigma\frac{\partial}{\partial\alpha}(\frac{\eta'}{\eta})] \\
&+ (b-\beta)[-2\omega\rho\frac{\partial\eta}{\partial\beta} - \rho\sigma\frac{\partial}{\partial\beta}(\frac{\eta'}{\eta})]
\end{aligned}
$$

$$(9) \qquad a' = 0 \, ,$$

$$(10) \qquad b' = 0 \, .$$

In these equations, we must make the substitutions

$$
\begin{aligned}
(11) \qquad &\frac{\partial\eta}{\partial\alpha} = 1, \qquad \frac{\partial}{\partial\alpha}(\frac{\eta'}{\eta}) = -\frac{\eta'}{\eta^2} \, , \\
&\frac{\partial\eta}{\partial\beta} = (x-1)^2 \, , \quad \frac{\partial}{\partial\beta}(\frac{\eta'}{\eta}) = 2\frac{x-1}{\eta} - \frac{\eta'}{\eta^2}(x-1)^2 \, .
\end{aligned}
$$

For each iteration of the successive approximation scheme, we produce numerically a particular vector solution $p(x)$ and two homogeneous vector solutions $h^1(x)$ and $h^2(x)$ of the system (7) – (10). We set the components of the reflection coefficients equal to a linear combination of the components of $p(x)$, $h^1(x)$, and $h^2(x)$,

$$
\begin{aligned}
&R_j^k = p_j(x) + a\,h_j^1(x) + b\,h_j^2(x) \, , \quad j = 1, 2, 3 \\
(12)\ldots \qquad &S_j^k = p_{j+3}(x) + a\,h_{j+3}^1(x) + b\,h_{j+3}^2(x) \, , \quad j = 1, 2, 3 \\
&a^k = p_7(x) + a\,h_7^1(x) + b\,h_7^2(x) = a \, ,
\end{aligned}
$$

$$(12) \qquad b^k = p_8(x) + a\, h_8^1(x) + b\, h_8^2(x) = b \; .$$

The multipliers a and b are given by the equations

$$\frac{\partial}{\partial a}\left\{\sum_{i=1}^{3}[(A_i - R_i^k(0))^2 + (B_i - S_i^k(0))^2]\right\} = 0 \; ,$$

(13)

$$\frac{\partial}{\partial b}\left\{\sum_{i=1}^{3}[(A_i - R_i^k(0))^2 + (B_i - S_i^k(0))^2]\right\} = 0 \; .$$

After making the substitutions (12), we obtain the values of a and b in the current approximation,

$$a = (f_1 e_{22} - f_2 e_{12}) \,/\, (e_{11}e_{22} - e_{12}e_{21}) \; ,$$

(14)

$$b = (e_{11}f_1 - e_{21}f_1) \,/\, (e_{11}e_{22} - e_{12}e_{21}) \; ,$$

where the right hand sides are given in terms of known quantities,

$$f_i = \sum_{\ell=1}^{3} h_\ell^i(0)(A_\ell - p_\ell(0)) + \sum_{\ell=1}^{3} h_{\ell+3}^i(0)(B_\ell - p_{\ell+3}(0))$$

$$(15) \quad e_{ij} = \sum_{\ell=1}^{6} h_\ell^i(0) h_\ell^j(0) \; , \quad j = 1,\, 2 \; ,$$

$$i = 1,\, 2 \; .$$

6. NUMERICAL EXPERIMENTS

Using the given data, and the initial approximation for refractive index $n(x) = 1.2 + 0.2(x-1)^2$, we determine the constants a and b in the function $n(x) = a + b(x-1)^2$ to one part in 10^6 after five iterations of quasilinearization. The successive approximations of the constants a

and b are listed in Table 1, labelled Trial 1, and the approximations of the index of refraction are shown in Fig. 6.

For the next experiment, we use data which are in error by $\pm 2\%$:

$$
(1) \qquad
\begin{aligned}
A_1 &= .134822 \times 10^{-2}, & B_1 &= .145461 \times 10^{-1}, \\
A_2 &= .316668 \times 10^{-3}, & B_2 &= .935364 \times 10^{-2}, \\
A_3 &= -.396321 \times 10^{-3}, & B_3 &= .601557 \times 10^{-2}.
\end{aligned}
$$

After five iterations, the initial approximation being the same as before, the constant a is found correct to within 0.3%, and b is correct to about 3%. On the other hand, the error in $n(x)$ ranges from 0.3% at $x = 1$ to only 0.7% at $x = 0$. The results are given in Table 1.

For each trial, the step length of integration is -.0025, and the integration scheme is Adams-Moulton. The time of calculations is 2 min. 12 sec. on the IBM 7044. The FORTRAN programs are found in Appendix G.

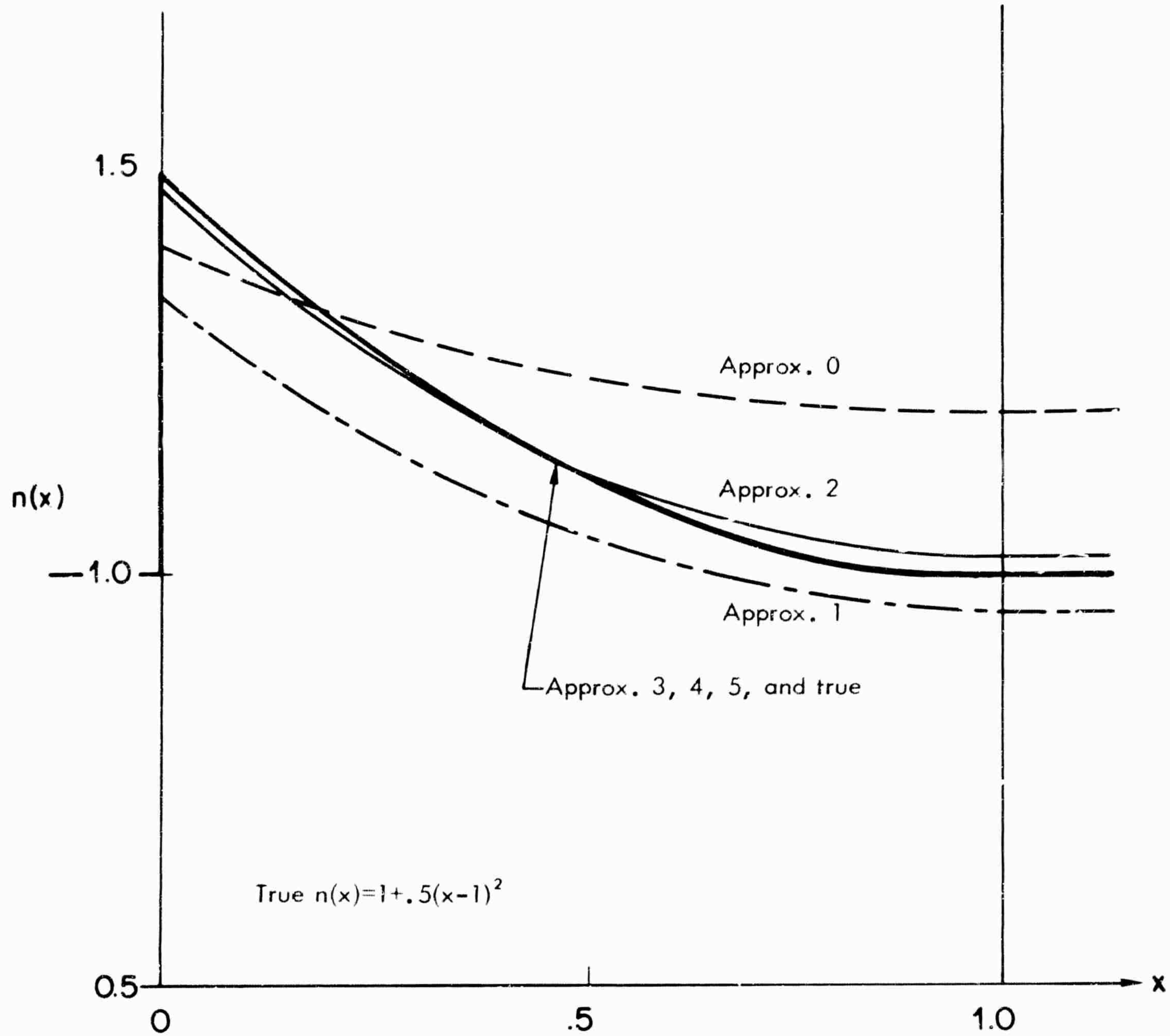


Fig. 6. Successive approximation of the index of refraction, Trial 1.

TABLE 1

TABLE 1

SUCCESSIVE APPROXIMATIONS OF THE PARAMETERS
a AND b IN THE EQUATION FOR REFRACTIVE INDEX

Approximation	Trial 1		Trial 2	
	a	b	a	b
0	1.2	0.2	1.2	0.2
1	0.9570026	0.3901124	0.9634062	0.3833728
2	1.0231246	0.4476300	1.0249423	0.4388065
3	1.0024801	0.4955928	1.0085360	0.4839149
4	0.9999989	0.4999477	1.0031811	0.4849384
5	0.9999993	0.4999996	1.0034835	0.4851507

7. DISCUSSION

Inverse problems in wave propagation, as well as in particle processes, can be computationally solved. The wave equation, being a partial differential equation, is replaced by a system of ordinary differential equations in one of several ways. In the previous chapter, we used Laplace transform methods. In this chapter, we assumed a solution of the form $u(x,t) = u(x)\, e^{-i\omega t}$, and we obtained ordinary differential equations for $u(x)$. Another Fourier decomposition might be

$$u(x,t) = \sum_{n=1}^{N} a_n(x) \sin nt \ ,$$

which results in second order ordinary differential equations for the functions $a_n(x)$. Another system of ordinary differential equations results when the space derivative is replaced by a finite difference,

$$\ddot{u}_n(t) \cong \frac{1}{c^2} \frac{u_{n+1}(t) - 2\, u_n(t) + u_{n-1}(t)}{\Delta^2}$$

These offer interesting possibilities for further studies.

REFERENCES

1. Lindsay, R. B., Mechanical Radiation, McGraw-Hill Book Company Inc., New York, 1960.

2. Morse, P. M., Vibration and Sound, McGraw-Hill Book Company, Inc., New York, 1948.

3. Osterberg, H., "Propagation of Plane Electromagnetic Waves in Inhomogeneous Media," J. Opt. Soc. America Vol. 48, pp. 513-521, 1958.

4. Schelkunoff, S. A., "Remarks Concerning Wave Propagation in Stratified Media," The Theory of Electromagnetic Waves, A Symposium, Interscience Publishers, Inc., New York, pp. 181-192, 1951.

5. Bellman, R., and R. Kalaba, "Functional Equations, Wave Propagation and Invariant Imbedding," J. Math. Mech. Vol. 8, pp. 683-704, 1959.

6. Blore, W. E., P. E. Robillard, and R. I. Primich, "35 and 70 Gc Phase-Locked CW Balanced-Bridge Model Measurement Radars," Microwave Journal, Vol. 7, 61-65, 1964.

7. Kyhl, R. L., "Directional Couplers," Technique of Microwave Measurements, edited by C. G. Montgomery, McGraw-Hill Book Company, Inc., New York, pp. 854-897, 1947.

8. Bellman, R., and R. Kalaba, Quasilinearization and Boundary Value Problems, American Elsevier Publishing Co., New York, 1965.

BLANK PAGE

CHAPTER EIGHT

DISCUSSION

Inverse problems have stimulated much interest in recent years, since the advent of modern electronic computers. The estimation of the structure of a complicated system, formerly unattainable by analytic means or by the use of a desk calculator, is now possible.

The determination of orbits from observations, is a kind of inverse problem in celestial mechanics, going back to Newton, Laplace, Gauss and others [1,2]. Ambarzumian [3], Borg [4], and others [5-8] considered the problem of determining a linear differential equation of Sturm-Liouville type given a spectrum of eigenvalues. The estimation of scattering potentials from the phase shift has been the concern of investigators in quantum theory [9-17]. Many inverse problems have been considered [18-45], especially in the fields of astrophysics, geophysics and geology. Some computational results have already been obtained for the structure of the earth's atmosphere and crust using actual geophysical data [19, 20, 36]. Some inverse problems fall within the domain of system identification, prediction and control [46-62], while others may be called design problems [63-66]. The common goal of all inverse problems is to determine the structure of a system which has a desired or observed characteristic output.

Computational procedures for the solution of inverse problems have been few and limited in scope. The methodologies put forth in this thesis may serve to widen the range of inverse problems which can now be solved. We formulate inverse problems as nonlinear boundary value problems, since we possess effective computational methods for solving many classes of nonlinear boundary value problems. These methods include quasilinearization, dynamic programming, invariant imbedding, and various combinations of these [67-69, 62]. A number of modifications of the basic techniques are given in Refs. 67, 70-72, describing more accurate solutions of linear algebraic equations, simultaneous calculations of successive approximations, and automatic evaluations of partial derivatives.

Much remains to be done to build a firm library of computational procedures for the solution of inverse problems. Both new and existing methods should be developed. In particular, system identification via invariant imbedding [51] appears promising.

REFERENCES

1. Gauss, K. F., Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections, Dover Publications, New York, 1963 (Translation of Theoria Motus, 1857).

2. Dubyago, A. D., The Determination of Orbits, The Macmillan Company, New York, 1961 (Translation by R. D. Burke, G. Gordon, L. N. Rowell, and F. T. Smith).

3. Ambarzumian, V. A., "Über eine Frage der Eigenwerttheorie," Zeitschrift für Physik, Vol. 53, pp. 690-695, 1929.

4. Borg, G., "Eine Umkehrung der Sturm-Liouvilleschen Eigenwertaufgabe. Bestimmung der Differentialgleichung durch die Eigenwerte," Acta Math., Vol. 78, pp. 1-96, 1946.

5. Gel'fand, I. M. and B. M. Levitan, "On the Determination of a Differential Equation from its Spectral Function," Izv. Akad. Nauk SSSR, Ser. Mat., Vol. 15, 309, 1951; Translations of the American Mathematical Society, Vol. 2, pp. 253-304, 1955.

6. Marcenko, V. A., "Certain Questions of the Theory of a Second-order Differential Operator," Dokl. Akad. Nauk SSSR, Vol. 72, pp. 457-460, 1950.

7. Krein, M. G., "Solution of the Inverse Sturm-Liouville Problem," Dokl. Akad. Nauk SSSR, Vol. 76, pp. 21-24, 1951.

8. Krein, M. G., "Determination of the Density of a Non-Homogeneous Symmetric Cord from its Frequency Spectrum," Dokl. Acad. Nauk SSSR, Vol. 76, pp. 345-348, 1951.

9. Bargmann, V., "On the Connection between Phase Shifts and Scattering Potential," Rev. Mod. Phys., Vol. 21, No. 3, pp. 488-493, 1949.

10. Levinson, N., "On the Uniqueness of the Potential in Schrödinger's Equation for a Given Asymptotic Phase," Danske Vid. Selsk. Math.-Fys. Medd., Vol. 25, 9, 1949.

11. Jost, R., and W. Kohn, "Equivalent Potentials," Phys. Rev., Vol. 88, pp. 382-385, 1952.

12. Kay, I. and H. E. Moses, "The Determination of the Scattering Potential from the Spectral Measure Function," Nuovo Cimento, Vol. X, No., 3, pp. 276-304, 1956.

13. Faddeyev, L. D., "The Inverse Problem in the Quantum Theory of Scattering." Uspekhi Matem. Nauk., Vol. 14. 57, 1959; J. Math. Phys. Vol. 4. No. 1. pp. 72-104. 1963.

14. Keller, J. B., "Determination of a Potential from its Energy Levels and Undectability of Quantization at High Energy," Amer. J. Phys., Vol. 30. pp. 22-26, 1962.

15. Calogero, F., "Maximum and Minimum Principle in Potential Scattering." II Nuovo Cimento, Vol. XXVIII, No. 2. pp. 1866-1879, 1963.

16. Hylleraas, E. A., "On the Inversion of Eigenvalue Problems," Annals of Physics. Vol. 25, pp. 309-324, 1963.

17. Hylleraas, E. A., "Determination of a Perturbing Potentia from its Scattering Phase Shift and Bound States Energy Levels," Nuclear Physics, Vol. 57, pp. 208-231, 1964.

18. Kaplan, L. D., "Inference of Atmospheric Structure from Remote Radiation Measurements," J. Opt. Soc. Am., Vol. 49, pp. 1004-1007, 1959.

19. Yamamoto, G., "Numerical Method for Estimating the Stratospheric Temperature Distribution from Satellite Measurements in the CO_2 Band," J. Meteor., Vol. 18, pp. 581-588, 1961.

20. Yamamoto, G., and A. Shimanuki, "The Determination of Lateral Diffusivity in Diabatic Conditions near the Ground from Diffusion Experiments," J. Atm. Sci., Vol. 21, pp. 187-196, 1964.

21. Plamer, E. P. and W. G. Zdunkowski. "Absolute Scattering Functions and Transmission Values for Interpreting Laser Light Scattering in the Mesosphere," J. Geophys. Res., Vol. 69, No. 11, pp. 2369-2377, 1964.

22. Karp, S. N. and J. Shmoys, Calculation of Charge Density Distribution of Multilayers from Transit Time Data, New York University, EM-82, July 1955.

23. Deirmendjian, D., "A Water Cloud Interpretation of Venus' Microwave Spectrum," Icarus, Vol. 3, No. 2, pp. 109-120 1964.

24. Horak, Henry G., "Diffuse Reflection by Planetary Atmospheres," Ap. J., Vol. 112, 445-463, 1950.

25. Sekera, Z., and J. V. Dave, "Determination of the Vertican Distribution of Ozone from the Measurement of Diffusely Reflected Ultra-Violet Solar Radiation," Planetary Space Sci., Vol. 5, pp. 122-136, 1961.

26. Goody, R., "The Transmission of Radiation Through an Inhomogeneous Atmosphere," J. of the Atmospheric Sciences, Vol. 21, No. 6, pp. 575–581, 1964.

27. Rozenberg, G. V., "Light Scattering in the Earth's Atmosphere," Soviet Physics Uspekhi, Vol. 3, No. 3, pp. 346–371, 1960.

28. Henyey, L. G., and J. L. Greenstein, "Diffuse Radiation in the Galaxy," Astrophys. Journal, Vol. 93, pp. 70–83, 1941.

29. Dave, J. V., "Meaning of Successive Iteration of the Auxiliary Equation in the Theory of Radiative Transfer," Astrophys. J., Vol. 140, No. 3, pp. 1292–1303, 1964.

30. King, J. I. F., "Inversion by Slabs of Varying Thickness," J. Atm. Sci., Vol. 21, pp. 324–326, 1964.

31. Sims, A. R., "Certain Aspects of the Inverse Scattering Problem," J. Soc. Indust. Appl. Math., Vol. 5, No. 4, pp. 183–205, 1957.

32. Kay, I., Remarks Concerning the Determination of Electron Distribution in the Ionosphere by Means of Radio Soundings, New York University, EM–149, November 1959.

33. Langer, R. E., "An Inverse Problem in Differential Equations," Bull. Am. Math. Soc., Vol. 39, pp. 814–820, October, 1933.

34. Alekseev, A. S., "Some Converse Problems in Wave Propagation Theory. I," Acad. Sci. USSR, Bull. Geophys. Ser., No. 11, pp. 944–948, 1962.

35. Alekseev, A. S., "Some Converse Problems in Wave Propagation Theory. II," Acad. Sci. USSR, Bull. Geophys. Ser., No. 11, pp. 949–954, 1962.

36. Dorman, J. and M. Ewing, "Numerical Inversion of Sesmic Surface Wave Dispersion Data and Crust–Mantle Structure in the New York–Pennsylvania Area," J. Geophys. Res., Vol. 67, No. 13, pp. 5227–5241, 1962.

37. Cannon, J. R., "Determination of Certain Parameters in Heat Conduction Problems," J. Math. Analys. Appl., Vol. 8, No. 2, pp. 188–201, 1964.

38. Tun, V., "The Theory of Heat Potential. I. Exact Values of Heat Potential and the Inverse Problem in the Theory of Heat Potential," Zh. Vychislit. Mat. Fiz. (USSR), Vol. 4, No. 4, pp. 660–670, 1964.

39. Weiss, G., "On the Inversion of the Specific-Heat Function," Progr. Theor. Phys., Vol. 22, No. 4, pp. 526-530, October 1959.

40. Harris, J. L., "Diffraction and Resolving Power," J. Opt. Soc. Am., Vol. 54, No. 7, pp. 931-936, July 1964.

41. Eskin, V. E., "Light Scattering as a Method of Studying Polymers," Soviet Physics Uspekhi, Vol. 7, No. 2, pp. 270-304, 1964.

42. Medgyessy, P., Decomposition of Superpositions of Distribution Functions, Akademiai Kiado, Budapest, 1961.

43. Bellman, R., C. Collier, H. Kagiwada, R. Kalaba and R. Selvester, "Estimation of Heart Parameters using Skin Potential Measurements," Comm. Assoc. for Computing Machinery, Vol. 7, pp. 666-668, 1964.

44. Bellman, R., H. Kagiwada, R. Kalaba, and S. Ueno, Inverse Problems in Radiative Transfer: Layered Media, The RAND Corporation, RM-4281-ARPA, December 1964.

45. Bellman, R., H. Kagiwada and R. Kalaba, "Orbit Determinationaas a Multipoint Boundary-Value Problem and Quasilinearization," Proc. Nat. Acad. Sci., Vol. 48, pp. 1327-1329, 1962.

46. Bellman, R., Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, New Jersey, 1961.

47. Bellman, R. and R. Kalaba, "An Inverse Problem in Dynamic Programming and Automatic Control," J. Math. Analys. Appl., Vol. 7, pp. 322-325, 1963.

48. Bellman, R., H. Kagiwada and R. Kalaba, Quasilinearization, System Identification, and Prediction, The RAND Corporation, RM-3812-PR, August 1963 (To appear in Proc. Soc. Eng. Sci.).

49. Bellman, R., H. Kagiwada, and R. Kalaba, "Quasilinearization, Boundary-Value Problems and Linear Programming," IEEE Transactions on Automatic Control, Vol. AC-10, No. 2, p. 199, April 1965.

50. Bellman, R., H. Kagiwada, and R. Kalaba, "Identification of Linear Systems via Numerical Inversion of Laplace Transforms," IEEE Transactions on Automatic Control, Vol, AC-10, No. 1, pp. 111-112, January 1965.

51. Bellman, R., H. Kagiwada, R. Kalaba, and R. Sridhar, Invariant Imbedding and Nonlinear Filtering Theory, The RAND Corporation, RM-4374-PR, December 1964 (To be published in the Journal of the Astronautical Sciences).

52. Bellman, R., H. Kagiwada and R. Kalaba, "On the Identification of Systems and the Unscrambling of Data - I: Hidden Periodicities," Proc. Nat. Acad. Sci., Vol. 53, pp. 907-910, May 1965.

53. Bellman, R., H. Kagiwada, and S. Ueno, "On the Identification of Systems and the Unscrambling of Data - II: An Inverse Problem in Radiative Transfer," Proc. Nat. Acad. Sci., Vol. 53, pp. 910-913, May 1965.

54. Bellman, R., B. Gluss and R. Roth. On the Identification of Systems and the Unscrambling of Data: Some Problems Suggested by Neurophysiology, The RAND Corporation, RM-4266-PR, October 1964.

55. Bellman, R., B. Gluss and R. Roth. Identification of Differential Systems with Time-Varying Coefficients, The RAND Corporation, RM-4288-PR, November 1964.

56. Kitamori, T., Automatic and Remote Control (Proc. 1st Internat. Cong. Internat. Federation Automatic Control, Moscow, 1960), Vol. II, pp. 613-618, Butterworth, London, 1961.

57. Wolf, A. A., and J. H. Dietz, "A Statistical Theory for Parameter Identification in Physical Systems," J. Franklin Institute, Vol. 274, pp. 369-400, 1962.

58. Kalinina, T. B. and F. M. Gol'tsman, "The Theory of Optimal Methods for Determining the Position of Model Bodies by Magnetometry in the Presence of Interference," Acad. Sci. USSR, Bull. Geophys. Ser., pp. 992-999, 1962.

59. Ho, Y. and B. H. Whalen, "An Approach to the Identification and Control of Linear Dynamic Systems with Unknown Parameters," IEEE Transactions on Automatic Control, Vol. AC-8, No. 3, July 1963.

60. Kushner, H. J., "A Simple Iterative Procedure for the Identification of the Unknown Parameters of a Linear Time Varying Discrete System," J. of Basic Engineering, pp. 227-235, June 1963.

61. Bellman, R., H. Kagiwada and R. Kalaba, "A Computational Procedure for Optimal System Design and Utilization," Proc. Nat. Acad. Sci., Vol. 48, pp. 1524-1528, 1962.

62. Bellman, R., H. Kagiwada and R. Kalaba, Numerical Studies of a Two-Point Nonlinear Boundary-Value Problem Using Dynamic Programming, Invariant Imbedding, and Quasilinearization, The RAND Corporation, RM-4069-PR, March 1964.

63. Tierney, M. S., P. Waltman, and G. M. Wing, Some Problems in the Optimal Design of Shields and Reflectors in Particle and Wave Physics, Sandia Laboratory, 1964.

64. Keller, J. B., The Inverse Scattering Problem in Geometrical Optics and the Design of Reflectors, New York University, EM-110, January 1958.

65. Salimov, R. B. and Ju. M. Molokovic, "On the Inverse Problem of Aerohydromechanics in Connection with Change of Profile," Izv. Vyss. Ucebn. Zaved. Matematica, Vol. 35, No. 4, pp. 150-157, 1963.

66. Stagg, G. W. and M. Watson, "Dynamic Programming - Transmission Line Design," IEEE International Convention Record, Part 3, pp. 55-61, 1964.

67. Bellman, R. and R. Kalaba, Quasilinearization and Boundary Value Problems, American Elsevier Publishing Co., New York, 1965.

68. Bellman, R. E., Dynamic Programming, Princeton University Press, Princeton, 1957.

69. Bellman, R., R. Kalaba, and G. M. Wing, "Invariant Imbedding and Mathematical Physics," J. Math. Phys., Vol. 1, pp. 280-308, 1960.

70. Bellman, R., R. Kalaba, and J. Lockett, Dynamic Programming and Ill-Conditioned Linear Systems - II., The RAND Corporation, RM-4201-PR, July 1964.

71. Bellman, R., "Successive Approximations and Computer Storage Problems in Ordinary Differential Equations," Communications of the ACM, Vol. 4, pp. 22-223, 1961.

72. Bellman, R., H. Kagiwada, and R. Kalaba, "Wengert's Numerical Method for Partial Derivatives, Orbit Determination, and Quasilinearization," Communications of the ACM, Vol. 8, pp. 231-232, April 1965.

APPENDICES

THE FORTRAN PROGRAMS

The library routines mentioned in these Appendices are

BET — Brooks, N. and G. R. Levesque, "RS W028 – Interval Test Routine, A MAP-Coded Subroutine Revised for use with 7040/7044 FORTRAN", RAND 7044 Library Routine W028, The RAND Corporation, Santa Monica, 1964.

INTS/INTM — Causey, R., W. L. Frank, W. L. Sibley, and F. Valadez, "RS W031 – RW INT, Adams-Moulton, Runge-Kutta Integration IBMAP Coded Subroutine (FORTRAN IV)", RAND 7044 Library Routine W031, The RAND Corporation, Santa Monica, 1964.

MATINV — Belcher, S., and B. S. Garbow, "RS W019 – ANF402 – Matrix Inversion with Accompanying Solution of Linear Equations", RAND 7044 Library Routine W019, The RAND Corporation, Santa Monica, 1964.

SIMPLX — Clasen, R. J., "RS MSUB – Linear Programming Subroutine, FORTRAN IV Coded", RAND 7044 Library Routine W009, The RAND Corporation, Santa Monica, 1964.

APPENDIX A

PROGRAMS FOR ORBIT DETERMINATION

PROGRAM A.1. PRODUCTION OF OBSERVATIONS

The complete program is listed:

MAIN program

DAUX subroutine

The following library routine is required:

INTS/INTM

```
$JOB             2890,3BODY,K 160,5,100,100,C
$IBJOB           MAP
$IBFTC MAIN      REF
      COMMON ALPHA,X1,Y1,C(4),T(51)
C
C            3 BODY ORBIT DETERMINATION
C
    1 READ(5,100)NPRNT,MPRNT,ALPHA,X1,Y1,DELTA
      WRITE(6,90)NPRNT,MPRNT,ALPHA,X1,Y1,DELTA
      READ(5,101)(C(I),I=1,4)
      WRITE(6,91)(C(I),I=1,4)
C
      T(2)=0.0
      T(3)=DELTA
      DO 2 L=4,7
    2 T(L)=C(L-3)
      CALL INTS(T,4,2,0,0,0,0,0,0)
      THETA=ATAN2(T(6),T(4)-1.0)
      SN=SIN(THETA)
      CS=COS(THETA)
      TN=SN/CS
      WRITE(6,92)
      WRITE(6,93)T(2),T(4),T(5),T(6),T(7),THETA,TN
C
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
    3 CALL INTM
      THETA=ATAN2(T(6),T(4)-1.0)
      SN=SIN(THETA)
      CS=COS(THETA)
      TN=SN/CS
    4 WRITE(6,93)T(2),T(4),T(5),T(6),T(7),THETA,TN
      GO TO 1
C
  100 FORMAT(2I12,4E12.8)
  101 FORMAT(6E12.8)
   90 FORMAT(1H12I20,4E20.8)
   91 FORMAT(1H06E20.8)
   92 FORMAT(///9X1HT,19X1HX,15X5HDX/DT,19X1HY,15X5HDY/DT,15X5HTHETA
     1,13X7HTANGENT//)
   93 FORMAT(F10.2,1P6E20.5)
      END
$IBFTC DAUX      REF
      SUBROUTINE DAUX
      COMMON ALPHA,X1,Y1,C(4),T(51)
C
      R= T(4)**2 + T(6)**2
      R=SQRT(R**3)
      R1=(X1-T(4))**2 + (Y1-T(6))**2
      R1=SQRT(R1**3)
      T(8)=T(5)
      T(9)=-T(4)/R + ALPHA*(X1-T(4))/R1
      T(10)=T(7)
      T(11)=-T(6)/R + ALPHA*(Y1-T(6))/R1
      RETURN
```

```
        END
$ENTRY          MAIN
            10          25          0.2          4.0          1.0          0.01
           2.0         0.0          0.0          0.5
```

59

PROGRAM A.2. DETERMINATION OF ORBIT

The complete program is listed:

MAIN program

INPUT subroutine

DAUX subroutine

FUN1 subroutine

FUN2 subroutine

PDR1 subroutine

PDR2 subroutine

START subroutine

The following library routines are required:

INTS/INTM

MATINV

```
$IBFTC MAIN     REF
      COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),W(4,251),ALPHA,
     1  H(5,5,251),P(5,251),A(50,50),B(50,1),X,U,Y,V,NPRNT,MPRNT,DTIME
      DIMENSION PIVOT(50),INDEX(50,2),IPIVOT(50)
C
C
C              THREE BODY ORBIT DETERMINATION
    1 CALL INPUT
      DO 8 I=1,5
      THET=THETA(I)
      ST=SIN(THET)
      CT=COS(THET)
      TN=ST/CT
    8 PRINT114,THET,TN
    2 CALL START
C                        K ITERATIONS
    3 DO 19 K=1,KMAX
      NEQ=30
    4 T(2)=0.0
      T(3)=HGRID
      DO 5 I=4,363
    5 T(I)=0.
      T(5)=1.0
      T(12)=1.0
      T(19)=1.0
      T(26)=1.0
      T(33)=1.0
      N=1
      X=W(1,1)
      Y=W(3,1)
    6 CALL INTS(T,NEQ,2,0,0,0,0,0,0)
        L=3
      DO 7 I=1,5
        L=L+1
      P(I,N)=T(L)
      DO 7 J=1,5
        L=L+1
    7 H(J,I,N)=T(L)
      PRINT49,T(2),((H(J,I,N),I=1,5),J=1,5)
C
C                 INTEGRATE OVER RANGE
C
      DO 11 M1=1,MPRNT
      DO 10 M2=1,NPRNT
        CALL INTM
      N=N+1
        X=W(1,N)
        Y=W(3,N)
C                STORE P'S AND H'S
        L=3
      DO 10 I=1,5
        L=L+1
      P(I,N)=T(L)
      DO 10 J=1,5
        L=L+1
```

```
   10  H(J,I,N)=T(L)
       PRINT49,T(2),((H(J,I,N),I=1,5),J=1,5)
   11  CONTINUE
C
C              COMPUTE CONSTANTS
       DO 14 I=1,5
         N=NGRID(I)
         THET=THETA(I)
       STHET=SIN(THET)
       CTHET=COS(THET)
       DO 13 J=1,5
   13    A(I,J)=H(J,1,N)*STHET -H(J,3,N)*CTHET
         B(I,1)=(1.-P(1,N))*STHET +P(3,N)*CTHET
   14  PRINT114,(A(I,JJ),JJ=1,5),B(I,1)
   15  CALL MATINV(A,5,B,1,DETERM,PIVOT,INDEX,IPIVOT)
       PRINT114,(B(I,1),I=1,5)
C
C              COMPUTE NEW W'S
       N=1
       DO 20 I=1,4
   20  W(I,N)=B(I,1)
       ALPHA=B(5,1)
       PRINT40,K,ALPHA
       TIME=0.0
       AT=ATAN2(W(3,N),W(1,N) - 1.0)
       TN=W(3,N)/(W(1,N)-1.0)
       PRINT50,TIME,(W(I,1),I=1,4),AT,TN
       DO 18 M1=1,MPRNT
       DO 17 M2=1,NPRNT
       N=N+1
         DO 17 I=1,4
         W(I,N)=P(I,N)
         DO 17 J=1,5
   17    W(I,N)=W(I,N) + B(J,1)*H(J,I,N)
       TIME=TIME+DTIME
       AT=ATAN2(W(3,N),W(1,N) - 1.0)
       TN=W(3,N)/(W(1,N)-1.0)
   18  PRINT50,TIME,(W(I,N),I=1,4),AT,TN
C
   19  CONTINUE
C
        GO TO 1
   40   FORMAT(1H0/40X 9HITERATION,I3,5X7HALPHA =,    E18.6//
      1                                    6X4H   T,14X1HX,19X2HX',18X1HY
      2                              19X2HY',15X5HANGLE,13X7HTANGENT)
   49  FORMAT(1H0F9.2,5E20.8/(10X5E20.8))
   50  FORMAT(F10.2,6E20.6)
  114  FORMAT(1H0  6E20.6)
          END
$IBFTC INPUT   REF
       SUBROUTINE INPUT
       COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),W(4,251),ALPHA,
      1  H(5,5,251),P(5,251),A(50,50),B(50,1),X,U,Y,V,NPRNT,MPRNT,DTIME
C
       READ110,NPRNT,MPRNT,KMAX
```

```
      PRINT10,NPRNT,MPRNT,KMAX
      READ111,HGRID,ALPHA
      PRINT11,HGRID,ALPHA
      F=NPRNT
      DTIME=F*HGRID
      READ120,(NGRID(I),THETA(I),I=1,5)
      PRINT20,(NGRID(I),THETA(I),I=1,5)
  110 FORMAT(6I12)
   10 FORMAT(1H06I20)
  111 FORMAT(6E12.8)
   11 FORMAT(1H06E20.8)
  120 FORMAT(I12,E12.8)
   20 FORMAT(I20,E20.8)
        RETURN
          END
$IBFTC DAUX     REF
      SUBROUTINE DAUX
C
      COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),W(4,251),ALPHA,
     1  H(5,5,251),P(5,251),A(50,50),B(50,1),X,U,Y,V,NPRNT,MPRNT,DTIM
     2  ,IFLAG
      DIMENSION XX(2),YY(2),ANS(2),PP(5),HH(5,5),PD(5),HD(5,5),PD1(3),
     1  PD2(3),AA(2)
C
      GO TO (10,20),IFLAG
C
   10 XX(1)=T(4)
      XX(2)=0.0
      YY(1)=T(6)
      YY(2)=0.0
      AA(1)=ALPHA
      AA(2)=0.0
      T(8)=T(5)
      CALL FUN1(XX,YY,AA,ANS)
      T(9)=ANS(1)
C
      T(10)=T(7)
      CALL FUN2(XX,YY,AA,ANS)
      T(11)=ANS(1)
      RETURN
C
   20 XX(1)=X
      XX(2)=0.0
      YY(1)=Y
      YY(2)=0.0
      AA(1)=ALPHA
      AA(2)=0.0
        L=3
      DO 1 I=1,5
        L=L+1
      PP(I)=T(L)
      DO 1 J=1,5
        L=L+1
    1 HH(J,I)=T(L)
C          DX/DT
```

```
      CALL FUN1(XX,YY,AA,ANS)
      CALL PDR1(XX,YY,AA,PD1)
      PD(1)=PP(2)
      PD(2)=ANS(1) + (PP(1)-X)*PD1(1) + (PP(3)-Y)*PD1(2)
     1 + (PP(5) - ALPHA)*PD1(3)
      DO 2 J=1,5
      HD(J,1)=HH(J,2)
    2 HD(J,2)=HH(J,1)*PD1(1) + HH(J,3)*PD1(2) + HH(J,5)*PD1(3)
C
C              DY/DT
      CALL FUN2(XX,YY,AA,ANS)
      CALL PDR2(XX,YY,AA,PD2)
      PD(3)=PP(4)
      PD(4)=ANS(1) + (PP(1)-X)*PD2(1) + (PP(3)-Y)*PD2(2)
     1 + (PP(5) - ALPHA)*PD2(3)
      DO 3 J=1,5
      HD(J,3)=HH(J,4)
    3 HD(J,4)=HH(J,1)*PD2(1) + HH(J,3)*PD2(2) + HH(J,5)*PD2(3.
C
      PD(5)=0.0
      DO 5 J=1,5
    5 HD(J,5)=0.0
C
      DO 4 I=1,5
        L=L+1
      T(L)=PD(I)
      DO 4 J=1,5
        L=L+1
    4 T(L)=HD(J,I)
      RETURN
      END
$IBFTC FUN1    REF
      SUBROUTINE FUN1(XX,YY,AA,ANS)
      DIMENSION XX(2),YY(2),AA(2),ANS(2)
      X=XX(1)
      Y=YY(1)
      A=AA(1)
      R13=(X**2 + Y**2)**1.5
      R23=((X-4.0)**2 + (Y-1.0)**2)**1.5
      ANS(1)=-X/R13 - A*(X-4.0)/R23
      RETURN
      END
$IBFTC FUN2    REF
      SUBROUTINE FUN2(XX,YY,AA,ANS)
      DIMENSION XX(2),YY(2),AA(2),ANS(2)
      X=XX(1)
      Y=YY(1)
      A=AA(1)
      R13=(X**2 + Y**2)**1.5
      R23=((X-4.0)**2 + (Y-1.0)**2)**1.5
      ANS(1)=-Y/R13 - A*(Y-1.0)/R23
      RETURN
      END
$IBFTC PDR1    REF
      SUBROUTINE PDR1(XX,YY,AA,PD1)
```

$
$ I
C

```
      DIMENSION XX(2),YY(2),AA(2),PD1(3)
      RR=(X-4.0)**2    + (Y-1.0)**2
      R25=RR**2.5
      X=XX(1)
      Y=YY(1)
      R13=RR**1.5
      A=AA(1)
      R15=RR**2.5
      RR=(X-4.0)**2    + (Y-1.0)**2
      R23=RR**1.5
      R25=RR**2.5
      PD1(1)=-1.0/R13 + 3.0*X**2/R15 - A/R23 + 3.0*A*(X-4.0)**2/R25
      PD1(2)=3.0*X*Y/R15 + 3.0*A*(X-4.0)*(Y-1.0)/R25
      PD1(3)=-(X-4.0)/R23
      RETURN
      END
$IBFTC PDR2    REF
      SUBROUTINE PDR2(XX,YY,AA,PD2)
      DIMENSION XX(2),YY(2),AA(2),PD2(3)
      X=XX(1)
      Y=YY(1)
      A=AA(1)
      RR=X**2 + Y**2
      R13=RR**1.5
      R23=RR**1.5
      RR=X**2 + Y**2
      R15=RR**2.5
      PD2(1)=3.0*X*Y/R15 + 3.0*A*(X-4.0)*(Y-1.0)/R25
      PD2(2)=-1.0/R13 + 3.0*Y**2/R15 - A/R23 + 3.0*A*(Y-1.0)**2/R25
      PD2(3)=-(Y-1.0)/R23
      RETURN
      END
$IBFTC START   REF
      SUBROUTINE START
      COMMON T(363),NEQ,KMAX,HGRID,NGRID(5),THETA(5),W(4,251),ALPHA,
     1  H(5,5,251),P(5,251),A(50,50),B(50,1),X,U,Y,V,NPRNT,MPRNT,DTIME
     2  ,IFLAG
C
      IFLAG=1
      K=0
      PRINT40,K
      N=1
      TIME=0.0
      T(2)=0.0
      T(3)=HGRID
      READ110,(T(I),I=4,7)
 110  FORMAT(6E12.8)
      CALL INTS(T,4,2,0,0,0,0,0,0)
      DO 3 I=1,4
    3 W(I,1)=T(I+3)
      PRINT50,TIME,(W(I,N),I=1,4)
      DO 2 M1=1,MPRNT
      DO 1 M2=1,NPRNT
      N=N+1
      CALL INTM
```

```
      DO 4 I=1,4
    4 W(I,N)=T(I+3)
    1 CONTINUE
      TIME=TIME+DTIME
    2 PRINT50,TIME,(W(I,N),I=1,4)
      IFLAG=2
        RETURN
C
   40   FORMAT(1H0/65X 9HITERATION,I3//
     1                                            26X4H    T,14X1HX,19X2HX',18X1HY,
     2                                   19X2HY')
   50 FORMAT(F30.2,4E20.6)
          END
```

APPENDIX B

PROGRAMS FOR RADIATIVE TRANSFER: LAYERED MEDIA

PROGRAM B.1. DETERMINATION OF c. THE THICKNESS OF THE LOWER LAYER

The complete program is listed:

MAIN program

DAUX subroutine

NONLIN subroutine

PANDH subroutine

LINEAR subroutine

OUTPUT-subroutine

ALBEDO subroutine

The following library routine is required:

INTS/INTM

```
$IBFTC RTINV
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
C
C           PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM / )
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1     1X11HTHICKNESS =, F10.4 /
     2     1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))   //
     3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C           PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
      CALL NONLIN
C
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
C           QUASILINEARIZATION ITERATIONS
```

```
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$IBFTC DAUX
      SUBROUTINE DAUX
      DIMENSION V2(7,7),X(3),F(7),G(7)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
      GO TO (1,2),IFLAG
C
CNONLINEAR
C
    1 L=3
      DO 4 I=1,N
      DO 4 J=1,I
      L=L+1
    4 V2(I,J)=T(L)
      DO 5 I=1,N
      DO 5 J=I,N
    5 V2(I,J)=V2(J,I)
      L=L+1
      VLAM2=T(L)
      SIG=T(2)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=VLAM2
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 6 I=1,N
      F(I)=0.0
      DO 7 K=1,N
    7 F(I)=F(I) + WR(K)*V2(I,K)
    6 F(I)=0.5*F(I) + 1.0
C
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
      DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
    8 T(L)=DR
      DO 9 I=1,1
      L=L+1
    9 T(L)=0.0
```

```
      RETURN
C
C
CLINEAR
C
    2 SIG=T(2)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 16 I=1,N
      F(I)=0.0
      DO 17 K=1,N
   17 F(I)=F(I) + WR(K)*R2(I,K)
   16 F(I)=0.5*F(I) + 1.0
C
CP'S
C
      L=3
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 V2(I,J)=T(L)
      DO 15 I=1,N
      DO 15 J=I,N
   15 V2(I,J)=V2(J,I)
      L=L+1
      VLAM2=T(L)
C
      DO 10 I=1,N
      G(I)=0.0
      DO 10 K=1,N
   10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
      ARG=10.0*(Y-XLAM(2))
      XTANX=-10.0*XLAM(1)*(1.0-(TANH(ARG))**2)
        M=3+NEQ
      DO 12 I=1,N
      DO 12 J=1,I
      FIJ=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
      T1=      -AR(I,J)*(V2(I,J)-R2(I,J))
      T2=      0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=      CAPF
      T4=(VLAM2-XLAM(2))*XTANX*FIJ
        M=M+1
   12 T(M)=T1+T2+T3+T4
      DO 19 I=1,1
        M=M+1
   19 T(M)=0.0
C
CH'S
C
```

```
      DO 100 K=1,1
C
      DO 24 I=1,N
      DO 24 J=1,I
      L=L+1
   24 V2(I,J)=T(L)
      DO 25 I=1,N
      DO 25 J=I,N
   25 V2(I,J)=V2(J,I)
      L=L+1
      VLAM2=T(L)
C
      DO 20 I=1,N
      G(I)=0.0
      DO 20 J=1,N
   20 G(I)=G(I) +   V2(I,J)*WR(J)
C
      DO 22 I=1,N
      DO 22 J=1,I
      FIJ=F(I)*F(J)
      T1=      -AR(I,J)*V2(I,J)
      T2=      0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=0.0
      T4=VLAM2*XTANX*FIJ
        M=M+1
   22 T(M)=T1+T2+T3+T4
C
      DO 29 I=1,1
        M=M+1
   29 T(M)=0.0
  100 CONTINUE
      RETURN
      END
$IBFTC NONLIN
      SUBROUTINE NONLIN
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPPNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
C           NONLINEAR D.E. FOR TRUE SOLUTION OR FOR INITIAL APPROX.
C
      IFLAG=1
      T(2)=0.0
      T(3)=DELTA
      M=1
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=0.0
      R(L1,M)=R2(I,J)
    1 T(L3)=R2(I,J)
      L3=L3+1
```

```
    2   T(L3)=XLAM(2)
C
        NEQ=(N*(N+1))/2 + 1
        CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
        SIG=T(2)
        CALL OUTPUT
C
        DO 5 M1=1,M1MAX
        DO 4 M2=1,NPRNT
        CALL INTM
        M=M+1
        L1=0
        L3=3
        DO 3 I=1,N
        DO 3 J=1,I
        L1=L1+1
        L3=L3+1
        R2(I,J)=T(L3)
    3   R(L1,M)=R2(I,J)
    4   SIG=T(2)
    5   CALL OUTPUT
C
        RETURN
        END
$IBFTC PANDH
        SUBROUTINE PANDH
        COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
       1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
       2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
       3  H2(7,7,3),CONST(3),NEQ
        IFLAG=2
        T(2)=0.0
        T(3)=DELTA
        M=1
C P'S
C
        L1=0
        L3=3
        DO 1 I=1,N
        DO 1 J=1,I
        L1=L1+1
        L3=L3+1
        P(L1,M)=0.0
    1   T(L3)=P(L1,M)
        L3=L3+1
        PLAM(2)=0.0
    2   T(L3)=PLAM(2)
C
C H'S
C
        DO 7 K=1,1
        L1=0
        DO 3 I=1,N
        DO 3 J=1,I
```

```
      L1=L1+1
      L3=L3+1
      H(L1,K,M)=0.0
    3 T(L3)=H(L1,K,M)
C
      L3=L3+1
    6 HLAM(2,K)=1.0
    7 T(L3)=HLAM(2,K)
C
      L=0
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
    8 R2(I,J)=R(L,M)
      DO 9 I=1,N
      DO 9 J=I,N
    9 R2(I,J)=R2(J,I)
C
      NEQ=2*((N*(N+1))/2 + 1)
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      LMAX=(N*(N+1))/2
      PRINT52,T(2),(P(L,M),H(L,1,M),L=1,LMAX)
   52 FORMAT(1H0F9.4,5E20.8/(10X5E20.8))
C
      DO 51 M1=1,M1MAX
      DO 50 M2=1,NPRNT
      CALL INTM
      M=M+1
CPREV.APPROX. R(I,J)
      L1=0
      DO 10 I=1,N
      DO 10 J=1,I
      L1=L1+1
   10 R2(I,J)=R(L1,M)
      DO 11 I=1,N
      DO 11 J=I,N
   11 R2(I,J)=R2(J,I)
      L1=0
      L3=3
      DO 12 I=1,N
      DO 12 J=1,I
      L1=L1+1
      L3=L3+1
   12 P(L1,M)=T(L3)
      L3=L3+1
      DO 13 K=1,1
      L1=0
      DO 14 I=1,N
      DO 14 J=1,I
      L1=L1+1
      L3=L3+1
   14 H(L1,K,M)=T(L3)
   13 L3=L3+1
   50 CONTINUE
   51 PRINT52,T(2),(P(L.M),H(L,1,M),L=1,LMAX)
```

```
      RETURN
      END
$IBFTC LINEAR
      SUBROUTINE LINEAR
      DIMENSION CHKI(3)
      DIMENSION A(49,3),B(49),EMAT(50,50),          PIVOT(50),INDEX(50,2,
     1,IPIVOT(50),FVEC(50,1)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,1
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
CLEAST SQUARES
      DO 5 K=1,1
      L=0
      DO 5 I=1,N
      DO 5 J=1,N
      L=L+1
    5 A(L,K)=H2(I,J,K)
      L=0
      DO 6 I=1,N
      DO 6 J=1,N
      L=L+1
    6 B(L)=B2(I,J) - P2(I,J)
C
      LMAX=N**2
      PRINT60
   60 FORMAT(1H0)
      DO 61 L=1,LMAX
   61 PRINT82,(A(L,K),K=1,1),B(L)
C
      DO 8 I=1,1
      DO 7 J=1,1
      SUM=0.0
      DO 9 L=1,LMAX
    9 SUM=SUM + A(L,I)*A(L,J)
    7 EMAT(I,J)=SUM
```

```
      SUM=0.0
      DO 10 L=1,LMAX
   10 SUM=SUM + A(L,I)*B(L)
    8 FVEC(I,1)=SUM
C
      PRINT60
      DO 81 I=1,1
   81 PRINT82,(EMAT(I,J),J=1,1),FVEC(I,1)
   82 FORMAT(10X6E20.8)
C
      FVEC(1,1)=FVEC(1,1)/EMAT(1,1)
C
      DO 11 I=1,1
   11 CONST(I)=FVEC(I,1)
C
      XLAM(2)=CONST(1)
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))  //
     3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
C
CNEW APPROXIMATION
C
      M=1
      L=0
      DO 12 I=1,N
      DO 12 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 13 K=1,1
   13 SUM =SUM + CONST(K)*H(L,K,M)
   12 R(L,M)=SUM
      L=0
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 R2(I,J)=R(L,M)
      SIG=0.0
      CALL OUTPUT
C
      DO 50 M1=1,M1MAX
      DO 18 M2=1,NPRNT
      M=M+1
      L=0
      DO 15 I=1,N
      DO 15 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 16 K=1,1
   16 SUM=SUM + CONST(K)*H(L,K,M)
   15 R(L,M)=SUM
      L=0
      DO 17 I=1,N
      DO 17 J=1,I
```

```
      L=L+1
   17 R2(I,J)=R(L,M)
   18 SIG=SIG + DELTA
   50 CALL OUTPUT
C
      RETURN
      END
$IBFTC OUTPUT
      SUBROUTINE OUTPUT
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      DO 1 I=1,N
      DO 1 J=I,N
    1 R2(I,J)=R2(J,I)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      PRINT100, SIG,Y,Z
  100 FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/)
      DO 2 J=1,N
    2 PRINT101,J,(R2(I,J),I=1,N)
  101 FORMAT(I10, 7F10.6)
      RETURN
      END
$IBFTC ALBEDO
      SUBROUTINE ALBEDO(Y,X,Z)
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      ARG=10.0*(Y-X(3))
      Z=X(1) + X(2)*TANH(ARG)
      RETURN
      END
```

480

PROGRAM B.2. DETERMINATION OF T, THE OVERALL OPTICAL THICKNESS

The complete program is listed:

MAIN program

DAUX subroutine

NONLIN subroutine

PANDH subroutine

LINEAR subroutine

OUTPUT subroutine

ALBEDO subroutine

The following library routine is required:

INTS/INTM

```
$IBFTC RTINV
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
C
C          PHASE I
C
    1 READ1000,N
      PRINT899
       RINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM / )
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))  //
     3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C          PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
      CALL NONLIN
C
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
C          QUASILINEARIZATION ITERATIONS
```

```
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$IBFTC DAUX
      SUBROUTINE DAUX
      DIMENSION V2(7,7),X(3),F(7),G(7)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
      GO TO (1,2),IFLAG
C
CNONLINEAR
C
    1 L=3
      DO 4 I=1,N
      DO 4 J=1,I
      L=L+1
    4 V2(I,J)=T(L)
      DO 5 I=1,N
      DO 5 J=I,N
    5 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
      SIG=T(2)
      Y=VTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 6 I=1,N
      F(I)=0.0
      DO 7 K=1,N
    7 F(I)=F(I) + WR(K)*V2(I,K)
    6 F(I)=0.5*F(I) + 1.0
C
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
      DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
    8 T(L)=DR*VTAU
      DO 9 I=1,1
      L=L+1
    9 T(L)=0.0
```

```
      RETURN
C
C
CLINEAR
C
    2 SIG=T(2)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 16 I=1,N
      F(I)=0.0
      DO 17 K=1,N
   17 F(I)=F(I) + WR(K)*R2(I,K)
   16 F(I)=0.5*F(I) + 1.0
C
CP'S
C
      L=3
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 V2(I,J)=T(L)
      DO 15 I=1,N
      DO 15 J=I,N
   15 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
C
      DO 10 I=1,N
      G(I)=0.0
      DO 10 K=1,N
   10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
      ARG=10.0*(Y-XLAM(2))
      PARTL=10.0*SIG*XLAM(1)*(1.0-(TANH(ARG))**2)
        M=3+NEQ
      DO 12 I=1,N
      DO 12 J=1,I
      FIJ=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
      T1=-XTAU*AR(I,J)*(V2(I,J)-R2(I,J))
      T2=XTAU*0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=XTAU*CAPF
      T4=(VTAU-XTAU)*(CAPF + XTAU*FIJ*PARTL)
        M=M+1
   12 T(M)=T1+T2+T3+T4
      DO 19 I=1,1
        M=M+1
   19 T(M)=0.0
C
CH'S
C
```

```
      DO 100 K=1,1
C
      DO 24 I=1,N
      DO 24 J=1,I
      L=L+1
   24 V2(I,J)=T(L)
      DO 25 I=1,N
      DO 25 J=I,N
   25 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
C
      DO 20 I=1,N
      G(I)=0.0
      DO 20 J=1,N
   20 G(I)=G(I) +  V2( ,J)*WR(J)
C
      DO 22 I=1,N
      DO 22 J=1,I
      FIJ=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
      T1=-XTAU*  (I,J)*V2(I,J)
      T2=XTAU*0.5*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=0.0
      T4=VTAU*(CAPF + XTAU*FIJ*PARTL)
        M=M+1
   22 T(M)=T1+T2+T3+T4
C
      DO 29 I=1,1
        M=M+1
   29 T(M)=0.0
  100 CONTINUE
      RETURN
      END
$IBFTC NONLIN
      SUBROUTINE NONLIN
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
C          NONLINEAR D.E. FOR TRUE SOLUTION OR FOR INITIAL APPROX.
C
      IFLAG=1
      T(2)=0.0
      T(3)=DELTA
      M=1
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=0.0
      R(L1,M)=R2(I,J)
    1 T(L3)=R2(I,J)
```

```
      L3=L3+1
    2 T(L3)=XTAU
C
      NEQ=(N*(N+1))/2 + 1
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
      SIG=T(2)
      CALL OUTPUT
C
      DO 5 M1=1,M1MAX
      DO 4 M2=1,NPRNT
      CALL INTM
      M=M+1
      L1=0
      L3=3
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=T(L3)
    3 R(L1,M)=R2(I,J)
    4 SIG=T(2)
    5 CALL OUTPUT
C
      RETURN
      END
$IBFTC LINEAR
      SUBROUTINE LINEAR
      DIMENSION CHKI(3)
      DIMENSION A(49,3),B(49),EMAT(50,50),          PIVOT(50),INDEX(50,
     1,IPIVOT(50),FVEC(50,1)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,1
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
CLEAST SQUARES
```

```
      DO 5 K=1,1
      L=0
      DO 5 I=1,N
      DO 5 J=1,N
      L=L+1
    5 A(L,K)=H2(I,J,K)
      L=0
      DO 6 I=1,N
      DO 6 J=1,N
      L=L+1
    6 B(L)=B2(I,J) - P2(I,J)
C
      LMAX=N**2
      PRINT60
   60 FORMAT(1H0)
      DO 61 L=1,LMAX
   61 PRINT82,(A(L,K),K=1,1),B(L)
C
      DO 8 I=1,1
      DO 7 J=1,1
      SUM=0.0
      DO 9 L=1,LMAX
    9 SUM=SUM + A(L,I)*A(L,J)
    7 EMAT(I,J)=SUM
      SUM=0.0
      DO 10 L=1,LMAX
   10 SUM=SUM + A(L,I)*B(L)
    8 FVEC(I,1)=SUM
C
      PRINT60
      DO 81 I=1,1
   81 PRINT82,(EMAT(I,J),J=1,1),FVEC(I,1)
   82 FORMAT(10X6E20.8)
C
      FVEC(1,1)=FVEC(1,1)/EMAT(1,1)
C
      DO 11 I=1,1
   11 CONST(I)=FVEC(I,1)
C
      XTAU   =CONST(1)
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, E16.8 /
     2    1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))  //
     3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
C
CNEW APPROXIMATION
C
      M=1
      L=0
      DO 12 I=1,N
      DO 12 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 13 K=1,1
```

```
 13   SUM =SUM + CONST(K)*H(L,K,M)
 12   R(L,M)=SUM
      L=0
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
 14   R2(I,J)=R(L,M)
      SIG=0.0
      CALL OUTPUT
C
      DO 50 M1=1,M1MAX
      DO 18 M2=1,NPRNT
      M=M+1
      L=0
      DO 15 I=1,N
      DO 15 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 16 K=1,1
 16   SUM=SUM + CONST(K)*H(L,K,M)
 15   R(L,M)=SUM
      L=0
      DO 17 I=1,N
      DO 17 J=1,I
      L=L+1
 17   R2(I,J)=R(L,M)
 18   SIG=SIG + DELTA
 50   CALL OUTPUT
C
      RETURN
      END
$IBFTC PANDH   LIST
      SUBROUTINE PANDH
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      IFLAG=2
      T(2)=0.0
      T(3)=DELTA
      M=1
C P'S
C
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      P(L1,M)=0.0
  1   T(L3)=P(L1,M)
      L3=L3+1
      PTAU=0.0
  2   T(L3)=PTAU
C
```

```
C H'S
C
      DO 7 K-1,1
      L1=0
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      H(L1,K,M)=0.0
    3 T(L3)=H(L1,K,M)
C
      L3=L3+1
    6 HTAU(K)=1.0
    7 T(L3)=HTAU(K)
C
      L=0
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
    8 R2(I,J)=R(L,M)
      DO 9 I=1,N
      DO 9 J=I,N
    9 R2(I,J)=R2(J,I)
C
      NEQ=2*((N*.N+1))/2 + 1)
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      LMAX=(N*(N+1))/2
C
      DO 51 M1=1,M1MAX
      DO 50 M2=1,NPRNT
      CALL INTM
      M=M+1
CPREV.APPROX. R(I,J)
      L1=0
      DO 10 I=1,N
      DO 10 J=1,I
      L1=L1+1
   10 R2(I,J)=R(L1,M)
      DO 11 I=1,N
      DO 11 J=I,N
   11 R2(I,J)=R2(J,I)
      L1=0
      L3=3
      DO 12 I=1,N
      DO 12 J=1,I
      L1=L1+1
      L3=L3+1
   12 P(L1,M)=T(L3)
      L3=L3+1
      DO 13 K-1,1
      L1=0
      DO 14 I=1,N
      DO 14 J=1,I
      L1=L1+1
      L3=L3+1
```

```
   14  H(L1,K,M)=T(L3)
   13  L3=L3+1
   50  CONTINUE
   51  CONTINUE
       RETURN
       END
$IBFTC OUTPUT
       SUBROUTINE OUTPUT
       DIMENSION X(3)
       COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
      1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
      2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
      3  H2(7,7,3),CONST(3),NEQ
       DO 1 I=1,N
       DO 1 J=I,N
    1  R2(I,J)=R2(J,I)
       Y=XTAU*SIG
       X(1)=ZERLAM
       X(2)=XLAM(1)
       X(3)=XLAM(2)
       CALL ALBEDO(Y,X,Z)
       PRINT100, SIG,Y,Z
  100  FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/)
       DO 2 J=1,N
    2  PRINT101,J,(R2(I,J),I=1,N)
  101  FORMAT(I10, 7F10.6)
       RETURN
       END
$IBFTC ALBEDO
       SUBROUTINE ALBEDO(Y,X,Z)
       DIMENSION X(3)
       COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
      1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
      2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
      3  H2(7,7,3),CONST(3),NEQ
       ARG=10.0*(Y-X(3))
       Z=X(1) + X(2)*TANH(ARG)
       RETURN
       END
```

PROGRAM B.3. DETERMINATION OF THE TWO ALBEDOS AND THE THICKNESS OF THE LOWER LAYER

The complete program is listed:

MAIN program

DAUX subroutine

NONLIN subroutine

PANDH subroutine

LINEAR subroutine

OUTPUT subroutine

ALBEDO subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$JOB            2609,STRAT3,HK0160,5,0,20,P
$PAUSE
$IBJOB STRAT2  MAP
$IBFTC RTINV
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),         B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
C
C            PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1.N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM / )
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,(XLAM(I),I=1,3)
      PRINT902
      PRINT903,XTAU,(XLAM(I),I=1,3)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))   //
     3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C            PHASE II
C
    4 READ1001,XTAU,(XLAM(I),I=1,3)
      K=0
      PRINT904,K
      PRINT903,XTAU,(XLAM(I),I=1,3)
C
      CALL NONLIN
C
```

```
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
C            QUASILINEARIZATION ITERATIONS
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$IBFTC DAUX    LIST
      SUBROUTINE DAUX
      DIMENSION V2(7,7),X(3),F(7),G(7)
     1  ,VLAM(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),        B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      GO TO (1,2),IFLAG
C
C  NLINEAR
C
    1 L=3
      DO 4 I=1,N
      DO 4 J=1,I
      L=L+1
    4 V2(I,J)=T(L)
      DO 5 I=1,N
      DO 5 J=I,N
    5 V2(I,J)=V2(J,I)
      DO 51 I=1,3
      L=L+1
   51 VLAM(I)=T(L)
      SIG=T(2)
      Y=XTAU*SIG
      DO 52 I=1,3
   52 X(I)=VLAM(I)
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 6 I=1,N
      F(I)=0.0
      DO 7 K=1,N
    7 F(I)=F(I) + WR(K)*V2(I,K)
    6 F(I)=0.5*F(I) + 1.0
C
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
      DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
```

```
    8  T(L)=DR
       DO 9 I=1,3
       L=L+1
    9  T(L)=0.0
       RETURN
C
C
CLINEAR
C
    2  SIG=T(2)
       Y=XTAU*SIG
       DO 21 I=1,3
   21  X(I)=XLAM(I)
       CALL ALBEDO(Y,X,Z)
       ZLAMDA=Z
C
       DO 16 I=1,N
       F(I)=0.0
       DO 17 K=1,N
   17  F(I)=F(I) + WR(K)*R2(I,K)
   16  F(I)=0.5*F(I) + 1.0
C
CP'S
C
       L=3
       DO 14 I=1,N
       DO 14 J=1,I
       L=L+1
   14  V2(I,J)=T(L)
       DO 15 I=1,N
       DO 15 J=I,N
   15  V2(I,J)=V2(J,I)
       DO 18 I=1,3
       L=L+1
   18  VLAM(I)=T(L)
C
       DO 10 I=1,N
       G(I)=0.0
       DO 10 K=1,N
   10  G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
       ARG=10.0*(Y-XLAM(3))
       TARG=TANH(ARG)
       XTANX=-10.0*XLAM(2)*(1.0-TARG**2)
         M=3+NEQ
       DO 12 I=1,N
       DO 12 J=1,I
       FIJ=F(I)*F(J)
       CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
       T1=CAPF
       T2=-AR(I,J)*(V2(I,J)-R2(I,J))
     1   + 0.5*ZLAMDA*(F(I)*G(J) + F(J)*G(I))
       T3=(VLAM(1)-XLAM(1))*FIJ
       T4=(VLAM(2)-XLAM(2))*TARG*FIJ
       T5=(VLAM(3)-XLAM(3))*XTANX*FIJ
         M=M+1
```

```
   12  T(M)=T1+T2+T3+T4+T5
       DO 19 I=1,3
         M=M+1
   19  T(M)=0.0
C
CH'S
C
       DO 100 K=1,3
C
       DO 24 I=1,N
       DO 24 J=1,I
       L=L+1
   24  V2(I,J)=T(L)
       DO 25 I=1,N
       DO 25 J=I,N
   25  V2(I,J)=V2(J,I)
       DO 26 I=1,3
       L=L+1
   26  VLAM(I)=T(L)
C
       DO 20 I=1,N
       G(I)=0.0
       DO 20 J=1,N
   20  G(I)=G(I) +  V2(I,J)*WR(J)
C
       DO 22 I=1,N
       DO 22 J=1,I
       FIJ=F(I)*F(J)
       T1=0.0
       T2=-AR(I,J)*V2(I,J) + 0.5*ZLAMDA*(F(I)*G(J) + F(J)*G(I))
       T3=VLAM(1)*FIJ
       T4=VLAM(2)*TARG*FIJ
       T5=VLAM(3)*XTANX*FIJ
         M=M+1
   22  T(M)=T1+T2+T3+T4+T5
C
       DO 29 I=1,3
         M=M+1
   29  T(M)=0.0
  100  CONTINUE
       RETURN
       END
$IBFTC NONLIN
       SUBROUTINE NONLIN
       COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),       B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
C            NONLINEAR D.E. FOR TRUE SOLUTION OR FOR INITIAL APPROX.
C
       IFLAG=1
       T(2)=0.0
       T(3)=DELTA
       M=1
       L1=0
```

```
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=0.0
      R(L1,M)=R2(I,J)
    1 T(L3)=R2(I,J)
      DO 2 I=1,3
      L3=L3+1
    2 T(L3)=XLAM(I)
C
      NEQ=(N*(N+1))/2 + 3
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
      SIG=T(2)
      CALL OUTPUT
C
      DO 5 M1=1,M1MAX
      DO 4 M2=1,NPRNT
      CALL INTM
      M=M+1
      L1=0
      L3=3
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=T(L3)
    3 R(L1,M)=R2(I,J)
    4 SIG=T(2)
    5 CALL OUTPUT
C
      RETURN
      END
$IBFTC PANDH
      SUBROUTINE PANDH
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),       B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      IFLAG=2
      T(2)=0.0
      T(3)=DELTA
      M=1
C P'S
C
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      P(L1,M)=0.0
    1 T(L3)=P(L1,M)
```

```
      DO 2 I=1,3
      L3=L3+1
      PLAM(I)=0.0
    2 T(L3)=PLAM(I)
C
C H'S
C
      DO 7 K=1,3
      L1=0
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      H(L1,K,M)=0.0
    3 T(L3)=H(L1,K,M)
C
      DO 7 I=1,3
      L3=L3+1
      HLAM(I,K)=0.0
      IF(I-K)7,6,7
    6 HLAM(I,K)=1.0
    7 T(L3)=HLAM(I,K)
C
      L=0
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
    8 R2(I,J)=R(L,M)
      DO 9 I=1,N
      DO 9 J=I,N
    9 R2(I,J)=R2(J,I)
C
      NEQ=4*((N*(N+1))/2 + 3)
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      LMAX=(N*(N+1))/2
      PRINT52,T(2),(P(L,M),H(L,1,M),L=1,LMAX)
   52 FORMAT(1HOF9.4,5E20.8/(10X5E20.8))
C
      DO 51 M1=1,M1MAX
      DO 50 M2=1,NPRNT
      CALL INTM
      M=M+1
CPREV.APPROX. R(I,J)
      L1=0
      DO 10 I=1,N
      DO 10 J=1,I
      L1=L1+1
   10 R2(I,J)=R(L1,M)
      DO 11 I=1,N
      DO 11 J=I,N
   11 R2(I,J)=R2(J,I)
      L1=0
      L3=3
      DO 12 I=1,N
      DO 12 J=1,I
```

```
      L1=L1+1
      L3=L3+1
   12 P(L1,M)=T(L3)
      L3=L3+3
      DO 13 K=1,3
      L1=0
      DO 14 I=1,N
      DO 14 J=1,I
      L1=L1+1
      L3=L3+1
   14 H(L1,K,M)=T(L3)
   13 L3=L3+3
   50 CONTINUE
   51 PRINT52,T(2),(P(L,M),H(L,1,M),L=1,LMAX)
      RETURN
      END
$IBFTC LINEAR
      SUBROUTINE LINEAR
      DIMENSION CHKI(3)
      DIMENSION A(49,3),B(49),EMAT(50,50),          PIVOT(50),INDEX(50,2)
     1,IPIVOT(50),FVEC(50,1)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1   XLAM(3),        B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2   P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3   H2(7,7,3),CONST(3),NEQ
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,3
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
CLEAST SQUARES
      DO 5 K=1,3
      L=0
      DO 5 I=1,N
      DO 5 J=1,N
      L=L+1
    5 A(L,K)=H2(I,J,K)
      L=0
      DO 6 I=1,N
      DO 6 J=1,N
      L=L+1
```

```
    6  B(L)=B2(I,J) - P2(I,J)
C
       LMAX=N**2
       PRINT60
   60  FORMAT(1H0)
       DO 61 L=1,LMAX
   61  PRINT82,(A(L,K),K=1,3),B(L)
C
       DO 8 I=1,3
       DO 7 J=1,3
       SUM=0.0
       DO 9 L=1,LMAX
    9  SUM=SUM + A(L,I)*A(L,J)
    7  EMAT(I,J)=SUM
       SUM=0.0
       DO 10 L=1,LMAX
   10  SUM=SUM + A(L,I)*B(L)
    8  FVEC(I,1)=SUM
C
       PRINT60
       DO 81 I=1,3
   81  PRINT82,(EMAT(I,J),J=1,3),FVEC(I,1)
   82  FORMAT(10X6E20.8)
C
       CALL MATINV(EMAT,3,FVEC,1,DETERM,PIVOT,INDEX,IPIVOT)
C
       DO 11 I=1,3
   11  CONST(I)=FVEC(I,1)
C
       DO 20 I=1,3
   20  XLAM(I)=CONST(I)
       PRINT903,XTAU,(XLAM(I),I=1,3)
  903  FORMAT(1H0/
      1  1X11HTHICKNESS =, E16.8 /
      2     1X11HALBEDO(X) =, 20HA + B*TANH(10*(X-C))  //
      3  1X3HA =, E16.8, 10X3HB =, E16.8, 10X3HC =, E16.8 //)
C
CNEW APPROXIMATION
C
       M=1
       L=0
       DO 12 I=1,N
       DO 12 J=1,I
       L=L+1
       SUM=P(L,M)
       DO 13 K=1,3
   13  SUM =SUM + CONST(K)*H(L,K,M)
   12  R(L,M)=SUM
       L=0
       DO 14 I=1,N
       DO 14 J=1,I
       L=L+1
   14  R2(I,J)=R(L,M)
       SIG=0.0
       CALL OUTPUT
```

```
C
      DO 50 M1=1,M1MAX
      DO 18 M2=1,NPRNT
      M=M+1
      L=0
      DO 15 I=1,N
      DO 15 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 16 K=1,3
   16 SUM=SUM + CONST(K)*H(L,K,M)
   15 R(L,M)=SUM
      L=0
      DO 17 I=1,N
      DO 17 J=1,I
      L=L+1
   17 R2(I,J)=R(L,M)
   18 SIG=SIG + DELTA
   50 CALL OUTPUT
C
      RETURN
      END
$IBFTC OUTPUT
      SUBROUTINE OUTPUT
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),        B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      DO 1 I=1,N
      DO 1 J=I,N
    1 R2(I,J)=R2(J,I)
      Y=XTAU*SIG
      DO 3 I=1,3
    3 X(I)=XLAM(I)
      CALL ALBEDO(Y,X,Z)
      PRINT100, SIG,Y,Z
  100 FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/)
      DO 2 J=1,N
    2 PRINT101,J,(R2(I,J),I=1,N)
  101 FORMAT(I10, 7F10.6)
      RETURN
      END
$IBFTC ALBEDO
      SUBROUTINE ALBEDO(Y,X,Z)
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  XLAM(3),        B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PLAM(3),HLAM(3,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
      ARG=10.0*(Y-X(3))
      Z=X(1) + X(2)*TANH(ARG)
      RETURN
      END
$ENTRY          RTINV
```

```
           7
25446046E-0112923441E-0029707742E-0050000000E 0070292258E 0087076559E 00       032
97455396E 00                                                                    033
64742484E-0113985269E-0019091502E-0020897958E-0019091502E-0013985269E-00      0032
64742484E-01                                                                   0033
          10          10          4
        0.01
         1.0         0.5         0.1         0.5
         1.0         0.6         .09         0.4
$IBSYS
```

505

APPENDIX C

PROGRAMS FOR RADIATIVE TRANSFER: NOISY OBSERVATIONS

PROGRAM C.1. MANY ACCURATE OBSERVATIONS FOR THE DETERMINATION OF ALBEDO

The complete program is listed:

MAIN program

DAUX subroutine

ALBEDO subroutine

PANDH subroutine

LINEAR subroutine

NONLIN subroutine

OUTPUT subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$IBFTC RTINV
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
C
C            PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
     1            47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION /
     2            47X27HUNKNOWN THICKNESS OF MEDIUM //)
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1     1X11HTHICKNESS =, F10.4 /
     2     1X11HALBEDO(X) =, F6.2,2H +, F6.2,3HX +, F6.2,4HX**2 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C            PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
      CALL NONLIN
C
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
```

```
C              QUASILINEARIZATION ITERATIONS
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$IBFTC DAUX
      SUBROUTINE DAUX
      DIMENSION V2(7,7),X(3),F(7),G(7)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
      GO TO (1,2),IFLAG
C
CNONLINEAR
C
    1 L=3
      DO 4 I=1,N
      DO 4 J=1,I
      L=L+1
    4 V2(I,J)=T(L)
      DO 5 I=1,N
      DO 5 J=I,N
    5 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
      L=L+1
      VLAM1=T(L)
      L=L+1
      VLAM2=T(L)
      SIG=T(2)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=VLAM1
      X(3)=VLAM2
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 6 I=1,N
      F(I)=0.0
      DO 7 K=1,N
    7 F(I)=F(I) + WR(K)*V2(I,K)
    6 F(I)=0.5*F(I) + 1.0
C
      DO 8 I=1,N
      DO 8 J=1,I
      L=L+1
```

```
      DR=-AR(I,J)*V2(I,J) + ZLAMDA*F(I)*F(J)
    8 T(L)=DR*VTAU
      DO 9 I=1,3
      L=L+1
    9 T(L)=0.0
      RETURN
C
C
CLINEAR
C
    2 SIG=T(2)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      ZLAMDA=Z
C
      DO 16 I=1,N
      F(I)=0.0
      DO 17 K=1,N
   17 F(I)=F(I) + WR(K)*R2(I,K)
   16 F(I)=0.5*F(I) + 1.0
C
CP'S
C
      L=3
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 V2(I,J)=T(L)
      DO 15 I=1,N
      DO 15 J=I,N
   15 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
      L=L+1
      VLAM1=T(L)
      L=L+1
      VLAM2=T(L)
C
      DO 10 I=1,N
      G(I)=0.0
      DO 10 K=1,N
   10 G(I)=G(I) + (V2(I,K)-R2(I,K))*WR(K)
        M=3+NEQ
      DO 12 I=1,N
      DO 12 J=1,I
      FIJ=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
      T1=-XTAU*AR(I,J)*(V2(I,J)-R2(I,J))
      T2=0.5*XTAU*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=VTAU*CAPF
      T4=(VTAU-XTAU)*(XLAM(1)*Y+2.0*XLAM(2)*Y**2)*FIJ
      PROD=XTAU*Y*FIJ
```

```
      T5=(VLAM1-XLAM(1))*PROD
      T6=(VLAM2-XLAM(2))*PROD*Y
        M=M+1
   12 T(M)=T1+T2+T3+T4+T5+T6
      DO 19 I=1,3
        M=M+1
   19 T(M)=0.0
C
CH'S
C
      DO 100 K=1,3
C
      DO 24 I=1,N
      DO 24 J=1,I
      L=L+1
   24 V2(I,J)=T(L)
      DO 25 I=1,N
      DO 25 J=I,N
   25 V2(I,J)=V2(J,I)
      L=L+1
      VTAU=T(L)
      L=L+1
      VLAM1=T(L)
      L=L+1
      VLAM2=T(L)
C
      DO 20 I=1,N
      G(I)=0.0
      DO 20 J=1,N
   20 G(I)=G(I) +  V2(I,J)*WR(J)
C
      DO 22 I=1,N
      DO 22 J=1,I
      FIJ=F(I)*F(J)
      CAPF=-AR(I,J)*R2(I,J) + ZLAMDA*FIJ
      T1=-XTAU*AR(I,J)*V2(I,J)
      T2=0.5*XTAU*ZLAMDA*(F(I)*G(J)+F(J)*G(I))
      T3=VTAU*CAPF
      T4=VTAU*(XLAM(1)*Y+2.0*XLAM(2)*Y**2)*FIJ
      PROD=XTAU*Y*FIJ
      T5=VLAM1*PROD
      T6=VLAM2*PROD*Y
        M=M+1
   22 T(M)=T1+T2+T3+T4+T5+T6
C
      DO 29 I=1,3
        M=M+1
   29 T(M)=0.0
  100 CONTINUE
      RETURN
      END
$IBFTC ALBEDO
      SUBROUTINE ALBEDO(Y,X,Z)
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
```

```
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
      Z=X(1) + X(2)*Y + X(3)*Y**2
      RETURN
      END
$IBFTC PANDH
      SUBROUTINE PANDH
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
      IFLAG=2
      T(2)=0.0
      T(3)=DELTA
      M=1
C P'S
C
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      P(L1,M)=0.0
    1 T(L3)=P(L1,M)
      L3=L3+1
      PTAU=0.0
      T(L3)=PTAU
      DO 2 I=1,2
      L3=L3+1
      PLAM(I)=0.0
    2 T(L3)=PLAM(I)
C
C H'S
C
      DO 7 K=1,3
      L1=0
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      H(L1,K,M)=0.0
    3 T(L3)=H(L1,K,M)
C
      I    3+1
      H    .K)=0.0
      IF(K-1)5,4,5
    4 HTAU(K)=1.0
    5 T(L3)=HTAU(K)
      DO 7 I=1,2
      L3=L3+1
      HLAM(I,K)=0.0
      IF(K-I-1)7,6,7
    6 HLAM(I,K)=1.0
```

```
    7   T(L3)=HLAM(I,K)
C
        L=0
        DO 8 I=1,N
        DO 8 J=1,I
        L=L+1
    8   R2(I,J)=R(L,M)
        DO 9 I=1,N
        DO 9 J=I,N
    9   R2(I,J)=R2(J,I)
C
        NEQ=4*((N*(N+1))/2 + 3)
        CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
        DO 50 M1=1,M1MAX
        DO 50 M2=1,NPRNT
        CALL INTM
        M=M+1
CPREV.APPROX. R(I,J)
        L1=0
        DO 10 I=1,N
        DO 10 J=1,I
        L1=L1+1
   10   R2(I,J)=R(L1,M)
        DO 11 I=1,N
        DO 11 J=I,N
   11   R2(I,J)=R2(J,I)
        L1=0
        L3=3
        DO 12 I=1,N
        DO 12 J=1,I
        L1=L1+1
        L3=L3+1
   12   P(L1,M)[illegible]L3)
        L3=L3+3
        DO 13 K=1,3
        L1=0
        DO 14 I=1,N
        DO 14 J=1,I
        L1=L1+1
        L3=L3+1
   14   H(L1,K,M)=T(L3)
   13   L3=L3+3
   50   CONTINUE
        RETURN
        END
$IBFTC LINEAR
        SUBROUTINE LINEAR
        DIMENSION CHKI(3)
        DIMENSION A(49,3),B(49),EMAT(50,50),          PIVOT(50),INDEX(50,2)
      1,IPIVOT(50),FVEC(50,1)
        COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
      1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
      2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
      3  H2(7,7,3),CONST(3),NEQ
```

```
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,3
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
CLEAST SQUARES
      DO 5 K=1,3
      L=0
      DO 5 I=1,N
      DO 5 J=1,N
      L=L+1
    5 A(L,K)=H2(I,J,K)
      L=0
      DO 6 I=1,N
      DO 6 J=1,N
      L=L+1
    6 B(L)=B2(I,J) - P2(I,J)
C
      LMAX=N**2
      PRINT60
   60 FORMAT(1H0)
      DO 61 L=1,LMAX
   61 PRINT82,(A(L,K),K=1,3),B(L)
C
      DO 8 I=1,3
      DO 7 J=1,3
      SUM=0.0
      DO 9 L=1,LMAX
    9 SUM=SUM + A(L,I)*A(L,J)
    7 EMAT(I,J)=SUM
      SUM=0.0
      DO 10 L=1,LMAX
   10 SUM=SUM + A(L,I)*B(L)
    8 FVEC(I,1)=SUM
C
      PRINT60
      DO 81 I=1,3
   81 PRINT82,(EMAT(I,J),J=1,3),FVEC(I,1)
   82 FORMAT(10X6E20.8)
C
C              SAVE FOR CHECKING
```

```
C
      DO 83 I=1,3
      DO 84 J=1,3
   84 A(I,J)=EMAT(I,J)
   83 B(I)=FVEC(I,1)
C
C
      CALL MATINV(EMAT,3,FVEC,1,DETERM,PIVOT,INDEX,IPIVOT)
C
      DO 11 I=1,3
   11 CONST(I)=FVEC(I,1)
C
C             CHECK MATRIX INVERSE
C
      PRINT60
      DO 71 I=1,3
      DO 70 J=1,3
      CHKI(J)=0.0
      DO 70 L=1,3
   70 CHKI(J)=CHKI(J) + EMAT(I,L)*A(L,J)
   71 PRINT82,(CHKI(J),J=1,3)
C
      DO 72 J=1,3
      CHKI(J)=0.0
      DO 72 L=1,3
   72 CHKI(J)=CHKI(J) + EMAT(J,L)*B(L)
      PRINT82,(CHKI(J),J=1,3)
C
      XTAU=CONST(1)
      XLAM(1)=CONST(2)
      XLAM(2)=CONST(3)
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X12HALBEDO(X) = , F6.2, 17H + C1*X + C2*X**2,
     3    2X3HC1=, E18.6, 2X3HC2=, E18.6//)
C
CNEW APPROXIMATION
C
      M=1
      L=0
      DO 12 I=1,N
      DO 12 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 13 K=1,3
   13 SUM =SUM + CONST(K)*H(L,K,M)
   12 R(L,M)=SUM
      L=0
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 R2(I,J)=R(L,M)
      SIG=0.0
      CALL OUTPUT
```

```
C
      DO 50 M1=1,M1MAX
      DO 18 M2=1,NPRNT
      M=M+1
      L=0
      DO 15 I=1,N
      DO 15 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 16 K=1,3
   16 SUM=SUM + CONST(K)*H(L,K,M)
   15 R(L,M)=SUM
      L=0
      DO 17 I=1,N
      DO 17 J=1,I
      L=L+1
   17 R2(I,J)=R(L,M)
   18 SIG=SIG + DELTA
   50 CALL OUTPUT
C
      RETURN
      END
$IBFTC NONLIN
      SUBROUTINE NONLIN
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
C           NONLINEAR D.E. FOR TRUE SOLUTION OR FOR INITIAL APPROX.
C
      IFLAG=1
      T(2)=0.0
      T(3)=DELTA
      M=1
      L1=0
      L3=3
      DO 1 I=1,N
      DO 1 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=0.0
      R(L1,M)=R2(I,J)
    1 T(L3)=R2(I,J)
      L3=L3+1
      T(L3)=XTAU
      DO 2 I=1,2
      L3=L3+1
    2 T(L3)=XLAM(I)
C
      NEQ=(N*(N+1))/2 + 3
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
      SIG=T(2)
      CALL OUTPUT
C
```

```
      DO 5 M1=1,M1MAX
      DO 4 M2=1,NPRNT
      CALL INTM
      M=M+1
      L1=0
      L3=3
      DO 3 I=1,N
      DO 3 J=1,I
      L1=L1+1
      L3=L3+1
      R2(I,J)=T(L3)
    3 R(L1,M)=R2(I,J)
    4 SIG=T(2)
    5 CALL OUTPUT
C
      RETURN
      END
$IBFTC OUTPUT
      SUBROUTINE OUTPUT
      DIMENSION X(3)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(5),NEQ
      DO 1 I=1,N
      DO 1 J=I,N
    1 R2(I,J)=R2(J,I)
      Y=XTAU*SIG
      X(1)=ZERLAM
      X(2)=XLAM(1)
      X(3)=XLAM(2)
      CALL ALBEDO(Y,X,Z)
      PRINT100, SIG,Y,Z
  100 FORMAT(1H0 7HSIGMA =,F6.2, 4X5HTAU =, F6.2, 4X8HALBEDO =,F6.2/)
      DO 2 J=1,N
    2 PRINT101,J,(R2(I,J),I=1,N)
  101 FORMAT(I10, 7F10.6)
      RETURN
      END
$ENTRY          RTINV
           7
25446046E-0112923441E-0029707742E-0050000000E 0070292258E 0087076559E 00      032
97455396E 00                                                                   033
64742484E-0113985269E-0019091502E-0020897958E-0019091502E-0013985269E-00     0032
64742484E-01                                                                  0033
          10            10            2
         0.01
          1.0           0.5           2.0          -2.0
          1.0           0.5           2.0          -2.0
```

PROGRAM C.2. OBSERVATIONS FOR ONLY ONE ANGLE OF INCIDENCE, FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

MAIN program

LINEAR subroutine

The following subroutines are required from Program C.1:

DAUX subroutine

ALBEDO subroutine

PANDH subroutine

NONLIN subroutine

OUTPUT subroutine

The following routines are required:

MATINV

INTS/INTM

```
$IBFTC RTINV    LIST
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
     4   ,NINC,JINC(7),NOBS
C
C           PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
     1            47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION /
     2            47X27HUNKNOWN THICKNESS OF MEDIUM //)
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, F6.2,2H +, F6.2,3HX +, F6.2,4HX**2 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C           PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
      READ1000,NINC
      PRINT900,NINC
      READ1000,(JINC(I),I=1,NINC)
```

```
      PRINT900,(JINC(I),I=1,NINC)

      NOBS=NINC*N
      PRINT900,NOBS
C
C
      CALL NONLIN
C
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
C           QUASILINEARIZATION ITERATIONS
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$IBFTC LINEAR  LIST
      SUBROUTINE LINEAR
      DIMENSION CHKI(3)
      DIMENSION A(49,3),B(49),EMAT(50,50),         PIVOT(50),INDEX(50,2)
     1,IPIVOT(50),FVEC(50,1)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
     4  ,NINC,JINC(7),NOBS
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,3
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
CLEAST SQUARES
      DO 5 K=1,3
      L=0
```

```
      DO 5 IN=1,NINC
      I=JINC(IN)
      DO 5 J=1,N
      L=L+1
    5 A(L,K)=H2(I,J,K)
      L=0
      DO 6 IN=1,NINC
      I=JINC(IN)
      DO 6 J=1,N
      L=L+1
    6 B(L)=B2(I,J) - P2(I,J)
C
      LMAX=N**2
      PRINT60
   60 FORMAT(1H0)
      DO 61 L=1,NOBS
   61 PRINT82,(A(L,K),K=1,3),B(L)
C
      DO 8 I=1,3
      DO 7 J=1,3
      SUM=0.0
      DO 9 L=1,NOBS
    9 SUM=SUM + A(L,I)*A(L,J)
    7 EMAT(I,J)=SUM
      SUM=0.0
      DO 10 L=1,NOBS
   10 SUM=SUM + A(L,I)*B(L)
    8 FVEC(I,1)=SUM
C
      PRINT60
      DO 81 I=1,3
   81 PRINT82,(EMAT(I,J),J=1,3),FVEC(I,1)
   82 FORMAT(10X6E20.8)
C
C              SAVE FOR CHECKING
C
      DO 83 I=1,3
      DO 84 J=1,3
   84 A(I,J)=EMAT(I,J)
   83 B(I)=FVEC(I,1)
C
C
      CALL MATINV(EMAT,3,FVEC,1,DETERM,PIVOT,INDEX,IPIVOT)
C
      DO 11 I=1,3
   11 CONST(I)=FVEC(I,1)
C
C              CHECK MATRIX INVERSE
C
      PRINT60
      DO 71 I=1,3
      DO 70 J=1,3
      CHKI(J)=0.0
      DO 70 L=1,3
   70 CHKI(J)=CHKI(J) + EMAT(I,L)*A(L,J)
```

```
   71   PRINT62,(CHKI(J),J=1,3)
C
        DO 72 J=1,3
        CHKI(J)=0.0
        DO 72 L=1,3
   72   CHKI(J)=CHKI(J) + EMAT(J,L)*B(L)
        PRINT82,(CHKI(J),J=1,3)
C
        XTAU=CONST(1)
        XLAM(1)=CONST(2)
        XLAM(2)=CONST(3)
        PRINT903,XTAU;ZERLAM,XLAM(1),XLAM(2)
  903   FORMAT(1H0/
     1     1X11HTHICKNESS =, E18.6 /
     2     1X12HALBEDO(X) = , F6.2, 17H + C1*X + C2*X**2,
     3     2X3HC1=, E18.6, 2X3HC2=, E18.6//)
C
CNEW APPROXIMATION
C
        M=1
        L=0
        DO 12 I=1,N
        DO 12 J=1,I
        L=L+1
        SUM=P(L,M)
        DO 13 K=1,3
   13   SUM =SUM + CONST(K)*H(L,K,M)
   12   R(L,M)=SUM
        L=0
        DO 14 I=1,N
        DO 14 J=1,I
        L=L+1
   14   R2(I,J)=R(L,M)
        SIG=0.0
        CALL OUTPUT
C
        DO 50 M1=1,M1MAX
        DO 18 M2=1,NPRNT
        M=M+1
        L=0
        DO 15 I=1,N
        DO 15 J=1,I
        L=L+1
        SUM=P(L,M)
        DO 16 K=1,3
   16   SUM=SUM + CONST(K)*H(L,K,M)
   15   R(L,M)=SUM
        L=0
        DO 17 I=1,N
        DO 17 J=1,I
        L=L+1
   17   R2(I,J)=R(L,M)
   18   SIG=SIG + DELTA
   50   CALL OUTPUT
C
```

```
RETURN
END
```

PROGRAM C.3. ERRORS IN THE OBSERVATIONS FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

MAIN program

The following subroutines are required from Program C.1:

DAUX subroutine

ALBEDO subroutine

PANDH subroutine

NONLIN subroutine

OUTPUT subroutine

The folliwng subroutine is required from Program C.2:

LINEAR subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$JOB            26..,RTI...,..,...., . . .
$IBJOB RTINV2   MAP
$IBFTC RTINV    LIST
      DIMENSION DERR(7,7),C2(7,7)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
     4 ,NINC,JINC(7),NOBS
C
C            PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
     1           47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION /
     2           47X27HUNKNOWN THICKNESS OF MEDIUM //)
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, F6.2,2H +, F6.2,3HX +, F6.2,4HX**2 //)
      CALL NONLIN
      DO 3 I=1,N
      DO 3 J=1,N
    3 B2(I,J)=R2(I,J)
C
C
C            PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
```

```
      READ1000,NINC

      PRINT900,NINC
      READ1000,(JINC(I),I=1,NINC)
      PRINT900,(JINC(I),I=1,NINC)
      NOBS=NINC*N
      PRINT900,NOBS
C
C
C                    READ ERRORS AS DECIMALS
C
      DO 6 I=1,NINC
      READ1001,(DERR(I,J),J=1,N)
    6 PRINT901,(DERR(I,J),J=1,N)
C
C                    STORE CORRECT OBSERVATIONS
C
      DO 7 I=1,N
      DO 7 J=1,N
    7 C2(I,J)=B2(I,J)
C
C                    CORRUPT OBSERVATIONS
C
      PRINT100
      DO 81 IN=1,NINC
      I=JINC(IN)
      DO 8 J=1,N
    8 B2(I,J)=B2(I,J)*(1.0+DERR(IN,J))
   81 PRINT101,I,(B2(I,J),J=1,N)
  100 FORMAT(1H0)
  101 FORMAT(I10,7F10.6)
C
C
      CALL NONLIN
C
  904 FORMAT(1H1   13HAPPROXIMATION, I3/ )
C
C              QUASILINEARIZATION ITERATIONS
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C                    RESTORE CORRECT OBSERVATIONS
C
      DO 9 I=1,N
      DO 9 J=1,N
    9 C2(I,J)=C2(I,J)
C
C
      GO TO 4
      END
```

```
$ENTRY          RTINV
          7
25446046E-0112923441E-0029707742E-00[illegible]E 0070292258E 0087076559E 00     032
97455396E 00
64742484E-0113965269E-0019091502E-002[illegible]E-0019091502E-0013985269E-00    00 2
64742484E-01
         10          10          3
       0.01
        1.0         0.5         2.0        -2.0
        1.0         0.5         2.0        -2.0
          1
          4
    -.003740    +.017960    -.054560    +.000400    -.024000    -.016980
    -.007740
$IBSYS
```

124

PROGRAM C.4. MINIMAX CRITERION FOR THE DETERMINATION OF ALBEDO

A partial program is listed:

LINEAR subroutine

The MAIN program is required from Program C.2.

The following subroutines are required from Program C.1:

DAUX subroutine

ALBEDO subroutine

PANDH subroutine

NONLIN subroutine

OUTPUT subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$JOB            2609,RTINV3,HK016[illegible]
$IBFTC LINEAR  LIST
      SUBROUTINE LINEAR
      DIMENSION INFIX(8),AS(22,35),BS(22),TOL(4),KOUT(7),ERR(8),JH(22),
     1  XS(22),PS(22),YS(22),KB(35),ES(22,22),ZS(35),A(7,3),B(7)
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1  ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2  P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3  H2(7,7,3),CONST(3),NEQ
     4  ,NINC,JINC(7),NOBS
C
C              USE LINEAR PROGRAMMING TO MINIMIZE MAXIMUM DEVIATION
C
CBOUNDARY CONDITIONS
      MLAST=NPRNT*M1MAX + 1
      DO 1 K=1,3
      L=0
      DO 2 I=1,N
      DO 2 J=1,I
      L=L+1
    2 H2(I,J,K)=H(L,K,MLAST)
      DO 1 I=1,N
      DO 1 J=I,N
    1 H2(I,J,K)=H2(J,I,K)
      L=0
      DO 3 I=1,N
      DO 3 J=1,I
      L=L+1
    3 P2(I,J)=P(L,MLAST)
      DO 4 I=1,N
      DO 4 J=I,N
    4 P2(I,J)=P2(J,I)
C
C
C              ZERO ALL AS, BS
C
      DO 7 I=1,22
      BS(I)=0.0
      DO 7 J=1,35
    7 AS(I,J)=0.0
C
C              COLUMNS 1 - 6
C
      IN=JINC(1)
      DO 8 I=2,8
      DO 8 K=1,3
      J1=2*K-1
      AS(I,J1)=H2(IN,I-1,K) / B2(IN,I-1)
      J2=J1+1
    8 AS(I,J2)=-AS(I,J1)
      DO 9 I=9,15
      I1=I-7
      DO 9 J=1,6
    9 AS(I,J)=-AS(I1,J)
C
```

```
      DO 6 I=2,8
    6 BS(I)=1.0 - P2(IN,I-1)/32(IN,I-1)
      DO 66 I=9,15
   66 BS(I)=-BS(I-7)
C
C             COLUMN 7
C
      AS(1,7)=1.0
      DO 10 I=16,22
   10 AS(I,7)=-1.0
      DO 91 I=1,22
   91 PRINT94,I,(AS(I,J),J=1,7),B(I)
   94 FORMAT(1H04XI5,8E15.6)
   90 FORMAT(1H04XI5.7E15.6/(10X7E15.6))
C
C             COLUMNS 8 - 28
C
      DO 11 J=8,28
      I=J-6
   11 AS(I,J)=1.0
      DO 92 I=1,22
   92 PRINT90,I,(AS(I,J),J=8,28
C
C             COLUMNS 29 - 35
C
      DO 19 J=29,35
      L=J-28
      I1=L+1
      I2=L+8
      I3=L+15
      AS(I1,J)=-1.0
      AS(I2,J)=-1.0
   19 AS(I3,J)=+1.0
      DO 93 I=1,22
   93 PRINT90,I,(AS(I,J),J=29,35)
C
C             INPUT TO SIMPLX (RAND LIBRARY ROUTINE W009)
C
      INFIX(1)=4
      INFIX(2)=35
      INFIX(3)=22
      INFIX(4)=22
      INFIX(5)=2
      INFIX(6)=1
      INFIX(7)=100
      INFIX(8)=0
      TOL(1)=1.0E-5
      TOL(2)=1.0E-5
      TOL(3)=1.0E-3
      TOL(4)=1.0E-10
      PRM=0.0
C
C             SIMPLX
C
      CALL SIMPLX(INFIX,AS,BS,TOL,PRM,KOUT,ERR,JH,XS,PS,YS,KB,ES)
```

```
C
C             OUTPUT FROM SIMPLX
C
      IF(KOUT(1)-3)20,21,20
C
   20 PRINT60,KOUT(1),KOUT(2)
   60 FORMAT(/5X6I20)
      CALL EXIT
C
   21 PRINT60,(KOUT(I),I=1,7)
      PRINT60
      PRINT61,(ERR(I),I=1,4)
   61 FORMAT(/5X6E20.6)
      PRINT60,(JH(I),I=1,22)
      PRINT61,(XS(I),I=1,22)
      PRINT61,(KB(I),I=1,35)
C
C             FIND Z'S
C
      MF=INFIX(5)
      M =INFIX(4)
      DO 22 I=MF,M
      J=JH(I)
      IF(J)22,22,23
   23 ZS(J)=XS(I)
   22 CONTINUE
C
      PRINT62,(J,ZS(J),J=1,35)
   62 FORMAT(/5XI5,E20.6)
C
      DO 24 I=1,3
      I1=2*I-1
      I2=I1+1
   24 CONST(I)=ZS(I1) - ZS(I2)
C
      PRINT63,ZS(7)
   63 FORMAT(1H04X20HMAXIMUM DEVIATION =, E15.6)
C
C
      XTAU=CONST(1)
      XLAM(1)=CONST(2)
      XLAM(2)=CONST(3)
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, E18.6 /
     2    1X12HALBEDO(X) = , F6.2, 17H + C1*X + C2*X**2,
     3    2X3HC1=, E18.6, 2X3HC2=, E18.6//)
C
CNEW APPROXIMATION
C
      M=1
      L=0
      DO 12 I=1,N
      DO 12 J=1,I
      L=L+1
```

```
      SUM=P(L,M)
      DO 13 K=1,3
   13 SUM =SUM + CONST(K)*H(L,K,M)
   12 R(L,M)=SUM
      L=0
      DO 14 I=1,N
      DO 14 J=1,I
      L=L+1
   14 R2(I,J)=R(L,M)
      SIG=0.0
      CALL OUTPUT
C
      DO 50 M1=1,M1MAX
      DO 18 M2=1,NPRNT
      M=M+1
      L=0
      DO 15 I=1,N
      DO 15 J=1,I
      L=L+1
      SUM=P(L,M)
      DO 16 K=1,3
   16 SUM=SUM + CONST(K)*H(L,K,M)
   15 R(L,M)=SUM
      L=0
      DO 17 I=1,N
      DO 17 J=1,I
      L=L+1
   17 R2(I,J)=R(L,M)
   18 SIG=SIG + DELTA
   50 CALL OUTPUT
C
      RETURN
      END
```

PROGRAM C.5. DESIGN OF A SLAB

A partial program is isted:

MAIN program

The following subroutines are required from Program C.1:

DAUX subroutine

ALBEDO subroutine

PANDH subroutine

NONLIN subroutine

OUTPUT subroutine

The following subroutine is required from Program C.2:

LINEAR subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$JOB            2609,RTINV4,K016[illegible]
$IBJOB RTINV4   MAP
$IBFTC RTINV    LIST
      COMMON N,RT(7),WT(7),WR(7),AR(7,7),NPRNT,M1MAX,KMAX,DELTA,XTAU,
     1 ZERLAM,XLAM(2),B2(7,7),R2(7,7),IFLAG,R(28,101),T(1491),SIG,
     2 P(28,101),H(28,3,101),PTAU,PLAM(2),HTAU(3),HLAM(2,3),P2(7,7),
     3 H2(7,7,3),CONST(3),NEQ
     4 ,NINC,JINC(7),NOBS
C
C           PHASE I
C
    1 READ1000,N
      PRINT899
      PRINT900,N
      READ1001,(RT(I),I=1,N)
      PRINT901,(RT(I),I=1,N)
      READ1001,(WT(I),I=1,N)
      PRINT901,(WT(I),I=1,N)
      DO 2 I=1,N
      WR(I)=WT(I)/RT(I)
      DO 2 J=1,N
    2 AR(I,J)= 1.0/RT(I) + 1.0/RT(J)
C
  899 FORMAT(1H146X36HRADIATIVE TRANSFER - INVERSE PROBLEM /
     1           47X33HUNKNOWN QUADRATIC ALBEDO FUNCTION /
     2           47X27HUNKNOWN THICKNESS OF MEDIUM //)
 1000 FORMAT(6I12)
  900 FORMAT(6I20)
 1001 FORMAT(6E12.8)
  901 FORMAT(6E20.8)
      READ1000,NPRNT,M1MAX,KMAX
      PRINT900,NPRNT,M1MAX,KMAX
      READ1001,DELTA
      PRINT901,DELTA
      READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      PRINT902
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
  902 FORMAT(1H123HPHASE I - TRUE SOLUTION /)
  903 FORMAT(1H0/
     1    1X11HTHICKNESS =, F10.4 /
     2    1X11HALBEDO(X) =, F6.2,2H +, F6.2,3HX +, F6.2,4HX**2 //)
C
C
C           PHASE II
C
    4 READ1001,XTAU,ZERLAM,XLAM(1),XLAM(2)
      K=0
      PRINT904,K
      PRINT903,XTAU,ZERLAM,XLAM(1),XLAM(2)
C
      READ1000,NINC
      PRINT900,NINC
      READ1000,(JINC(I),I=1,NINC)
      PRINT900,(JINC(I),I=1,NINC)
      NOBS=NINC*N
```

```
      PRINT900,NOBS
C
      DO 6 I=1,NINC
      J=JINC(I)
      READ1001,(B2(J,K),K=1,N)
    6 PRINT901,(B2(J,K),K=1,N)
C
      CALL NONLIN
C
  904 FORMAT(1H1    13HAPPROXIMATION, I3/ )
C
C             QUASILINEARIZATION ITERATIONS
C
      DO 5 K1=1,KMAX
      PRINT904,K1
      CALL PANDH
      CALL LINEAR
    5 CONTINUE
C
C
C
      READ1000,IGO
      GO TO (1,4),IGO
      END
$ENTRY          RTINV
         7
25446046E-0112923441E-0029707742E-0050000000E 0070292258E 0087076559E 00     032
97455396E 00
64742484E-0113985269E-0019091502E-0020897958E-0019091502E-0013985269E-00     00 2
64742484E-01
        10             10             5
       .01
       1.0            0.5           2.0           -2.0
       1.0            0.5           2.0           -2.0
         1
         7
       .028           .144          .333          .505          .621          .689
       .722
$IBSYS          ENDJOB
```

APPENDIX D

PROGRAM FOR RADIATIVE TRANSFER:

ANISOTROPIC SCATTERING

PROGRAM D.1. PROGRAM FOR THE CALCULATION OF REFLECTED INTENSITIES

The complete program is listed:

MAIN program

LGNDRP subroutine

CTAU subroutine

DAUX subroutine

DCTNRY subroutine

SSTART subroutine

OUTPUT subroutine

The following library routine is required:

INTS/INTM

```
$JOB            2609,REFLX,HK0160,8M,1000C,50P,C
$IBJOB REFLX    MAP
$IBFTC MAIN     REF
C                    RADIATIVE TRANSFER MAIN PROGRAM                    ANIS0030
      COMMON     T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),       ANIS0040
     1 P(11,11,10),PW(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),     ANIS0050
     2 FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),  ANIS0060
     3 A(10,10),DELPHI(20),THETA(10)                                   ANIS0070
     4 ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,TAUONE,TAUTWO,
     5  DELTAU,OMEGA,QLBEDO,NEQ,NPHI,FLUX,MPRNT
C                                                                       ANIS0210
C                           RADIATIVE TRANSFER                          ANIS0220
C            DIFFUSE REFLECTION FROM A TWO-DIMENSIONAL FLAT LAYER       ANIS0230
C                                                                       ANIS0240
C     INTEGRATION OF SCATTERING COEFFICIENTS  S(M,K,L)                  ANIS0250
C     TAU IS THE INDEPENDENT VARIABLE                                   ANIS0260
C     INTENSITY IS COMPUTED FROM THE  S  COEFFICIENTS                   ANIS0270
C                                                                       ANIS0280
C       VARIABLES                  DEFINITIONS                          ANIS0290
C     TAU            OPTICAL THICKNESS, IN MEAN FREE PATHS              ANIS0300
C     S(M,K,L)       M-TH SCATTERING COMPONENT FOR MU=XL(K,NQUAD) AND   ANIS0310
C                      MU-ZERO = XL(L,NQUAD).                           ANIS0320
C     XL(K,NQUAD)    K-TH ROOT OF NQUAD-DEGREE LEGENDRE POLYNOMIAL      ANIS0330
C     WT(K,NQUAD)    CORRESPONDING CHRISTOFFEL WEIGHT                   ANIS0340
C                      BOTH XL AND WT ON INTERVAL 0 TO 1                ANIS0350
C     P(M,I,K)       I-TH DEGREE, (M-1)TH ORDER ASSOCIATED LEGENDRE     ANIS0360
C                      FUNCTION EVALUATED AT X=XL(K,NQUAD)              ANIS0370
C     ZINT(J,K,L)    SCATTERED INTENSITY FOR MU=XL(K,NQUAD), MU-ZERO =  ANIS0380
C                      XL(L,NQUAD), AND DELPHI(J)                       ANIS0390
C     DELPHI(J)      J-TH AZIMUTH ANGLE.  NPHI ANGLES ARE INPUT (DEGREES).ANIS0400
C                      DELPHI(J)=0 MEANS FORWARD DIRECTION.             ANIS0410
C                      DELPHI(J)=180 MEANS BACKWARD DIRECTION.          ANIS0420
C     THETA(K)       POLAR ANGLE OF OUTPUT.  THETA(K)=ARC COSINE(MU),   ANIS0430
C                      WHERE MU=XL(K,NQUAD).                            ANIS0440
C     C(I)           I-TH FOURIER COEFFICIENT IN EXPANSION OF PHASE     ANIS0450
C                      FUNCTION                                         ANIS0460
C     OMEGA          ALBEDO OF SINGLE SCATTERING                        ANIS0470
C     QLBEDO         ALBEDO OF EARTH'S SURFACE                          ANIS0480
C                                                                       ANIS0490
C                                                                       ANIS0500
C       CONSTANTS                  DEFINITIONS                          ANIS0510
C         NQUAD       DEGREE OF GAUSSIAN QUADRATURE                     ANIS0520
C         MMAX-1      DEGREE OF FOURIER EXPANSION                       ANIS0530
C         NEQ         NUMBER OF DIFFERENTIAL EQUATIONS                  ANIS0540
C                       NEQ=MMAX*NQUAD*(NQUAD+1)/2                      ANIS0550
C         N1          INTEGRATION OPTION WORD                           ANIS0560
C         NPRNT       NUMBER OF INTEGRATIONS PER PRINT INTERVAL         ANIS0570
C         TAUONE      INITIAL TAU                                       ANIS0580
C         TAUTWO      FINAL TAU                                         ANIS0590
C         DELTAU      INTEGRATING GRID SIZE                             ANIS0600
C         FLUX        INCIDENT FLUX / PI.                               ANIS0610
C                                                                       ANIS0620
C                                                                       ANIS0630
C       FLAGS                        MEANINGS                           ANIS0640
C         NFLAG=1     NQUAD AND MMAX FOR THIS PROBLEM ARE NOT THE SAME  ANIS0650
```

```
C                    AS IN THE PREVIOUS PROBLEM, SO THAT CERTAIN
C                    VARIABLES MUST BE EVALUATED AGAIN.
C        NFLAG=2    OTHERWISE
C
C        KFLAG=1    C(I) ARE ALL CONSTANT
C        KFLAG=2    C(I) ARE FUNCTIONS OF TAU
C
C        LFLAG=1    OMEGA IS CONSTANT
C        LFLAG=2    OMEGA IS A FUNCTION OF TAU
C
C
C     DICTIONARY ARRAYS
C        FACT, SIGN, DEL
C
C
C
C             INPUT LEGENDRE ROOTS AND CHRISTOFFEL WEIGHTS
C             SET UP A DICTIONARY OF CONSTANT COEFFICIENTS
C
      CALL DCTNRY
C
C             INPUT PROBLEM CONSTANTS
    9 READ 1003,NQUAD, MMAX, NFLAG, KFLAG, LFLAG, NPRNT, N1,MPRNT
        IF(NQUAD-2)9999,9999,10
   10 READ 1004,TAUONE,TAUTWO,DELTAU,OMEGA,QLBEDO
      READ 1004,(C(I),I=1,MMAX)
        NEQ=MMAX*(NQUAD*(NQUAD+1))/2
      READ 1006,FLUX,NPHI,(DELPHI(J),J=1,NPHI)
      PRINT 2001,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,NEQ,MPRNT
      PRINT 2002,TAUONE,TAUTWO,DELTAU,OMEGA,QLBEDO
      PRINT 2003,(C(I),I=1,MMAX)
      PRINT 2004,FLUX,NPHI,(DELPHI(J),J=1,NPHI)
C
      GO TO (16,17),NFLAG
C             NEW ASSOCIATED LEGENDRE POLYNOMIALS AND OTHER VARIABLES
   16 CALL LGNDRP
      DO 18 J=1,NQUAD
        WX=WT(J,NQUAD)/XL(J,NQUAD)
      DO 18 M=1,MMAX
      DO 18 I=M,MMAX
   18 PW(M,I,J)=P(M,I,J)*WX
C
      DO 19 K=1,NQUAD
      XI=1.0/XL(K,NQUAD)
      DO 19 L=1,K
      A(L,K)=XI+1.0/XL(L,NQUAD)
   19 A(K,L)=A(L,K)
C
   17 GO TO (20,21),KFLAG
C                     THE COEFFICIENTS, C, AND THUS K, ARE CONSTANTS
   20 DO 22 M=1,MMAX
        DO 22 I=M,MMAX
        JO=I-M+1
        JT=I+M-1
        J=I+M-2+1
```

```
   22 CK(I,M)=C(I)*FACT(JC,JT)*SGN(J)                                    ANIS1210
C                                                                        ANIS1220
C           INITIAL INTEGRATING STEP                                     ANIS1230
   21 CALL SSTART                                                        ANIS1240
C                                                                        ANIS1250
      CALL INTS(T,NEQ,N1,0,0,0,0,0,0)
C                                                                        ANIS1270
        J=3                                                              ANIS1280
      DO 25 M=1,MMAX                                                     ANIS1290
      DO 25 K=1,NQUAD                                                    ANIS1300
      DO 25 L=1,K                                                        ANIS1310
        J=J+1                                                            ANIS1320
      S(M,K,L)=T(J)                                                      ANIS1330
      Q(M,K,L)=S(M,K,L)                                                  ANIS1340
   25 S(M,L,K)=S(M,K,L)                                                  ANIS1350
C                                                                        ANIS1360
C          COMPUTE INTENSITIES AND OUTPUT                                ANIS1370
      CALL OUTPUT                                                        ANIS1380
C                                                                        ANIS1390
C          GENERAL INTEGRATING STEPS                                     ANIS1400
      DO 31 M1=1,MPRNT
   27 DO 26 N=1,NPRNT                                                    ANIS141
      CALL INTM                                                          ANIS1
        J=3                                                              ANIS1430
      DO 26 M=1,MMAX                                                     ANIS1440
      DO 26 K=1,NQUAD                                                    ANIS1450
      DO 26 L=1,K                                                        ANIS1460
        J=J+1                                                            ANIS1470
      S(M,K,L)=T(J)                                                      ANIS1480
   26 S(M,L,K)=S(M,K,L)                                                  ANIS1490
      CALL OUTPUT                                                        ANIS1500
C                                                                        ANIS1510
   28 DO 29 M=1,MMAX                                                     ANIS1530
        DO 29 K=1,NQUAD                                                  ANIS1540
        DO 29 L=1,K                                                      ANIS1550
        G=S(M,K,L)-Q(M,K,L)                                              ANIS1560
      IF(ABS(G)-.000005 )29,29,30
   29 CONTINUE                                                           ANIS1580
      GO TO 9                                                            ANIS1590
   30 DO 31 M=1,MMAX                                                     ANIS1600
      DO 31 K=1,NQUAD                                                    ANIS1610
      DO 31 L=1,K                                                        ANIS1620
   31 Q(M,K,L)=S(M,K,L)                                                  ANIS1630
C                                                                        ANIS1650
 9999 CALL EXIT                                                          ANIS1660
C                                                                        ANIS1670
 1003 FORMAT(12I6)                                                       ANIS1680
 1004 FORMAT(6E12.6)                                                     ANIS1690
 1006 FORMAT(E10.6,I10,5E10.6/(7E10.6))                                  ANIS1700
 2001 FORMAT(1H149X18HRADIATIVE TRANSFER///                              ANIS1710
     1 26XI4,46H-POINT GAUSSIAN QUADRATURE                         /     ANIS1720
     2 26XI4,46H-TERM EXPANSION OF PHASE FUNCTION                  /     ANIS1730
     3 30X,    7H NFLAG=,I1, 9H.  KFLAG=,I1, 9H.  LFLAG=,I1,1H.   /     ANIS1740
     4 26XI4,46H INTEGRATIONS PER PRINT INTERVAL                   /     ANIS1750
     5 26XI4,46H=INTEGRATION OPTION WORD                           /     ANIS1760
```

```
     6  26X14,46H DIFFERENTIAL EQUATIONS                                /          ANIS1770
     7  26X14, 6H MPRNT / )
 2002 FORMAT(30X28H INTEGRATION RANGE IN TAU IS,F9.4, 3H TO,F9.4/                 ANIS1780
     1         30X13H GRID SIZE IS, F7.4/                                          ANIS1790
     2         30X31H ALBEDO OF SINGLE SCATTERING IS, F7.4/                        ANIS1800
     3         30X29H ALBEDO OF EARTH'S SURFACE IS, F7.4 )                         ANIS1810
 2003 FORMAT(30X36H COEFFICIENTS IN PHASE EXPANSION ARE /                         ANIS1820
     1  (33X6F9.4))                                                                ANIS1830
 2004 FORMAT(30X17H INCIDENT FLUX IS ,F9.4/                                        ANIS1840
     1  26X14,46H DELTA PHI ANGLES ARE                                  /          ANIS1850
     2  (33X6F9.4))                                                                ANIS1860
        END                                                                        ANIS1870
$IBFTC LGNDRP  REF                                                                 LGND0010
      SUBROUTINE LGNDRP                                                            LGND0030
CASSOCF                      ASSOCIATED FUNCTIONS              8-21-62            LGND0040
      COMMON     T(7263),S(11,10,10),G(11,10,10),ZINT(20,10,10),
     1  P(11,11,10),PX(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),               LGND0060
     2  FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),
     3  A(10,10),DELPHI(20),THETA(10)                                              LGND0080
     4  ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,TAUONE,TAUTWO,
     5   DELTAU,OMEGA,GLBEDO,NEG,NPHI,FLUX,MPRNT
C                                                                                  LGND0220
C                                                                                  LGND0230
      DO 100 K=1,NQUAD                                                             LGND0240
      X=XL(K,NQUAD)                                                                LGND0250
      XX=X**2                                                                      LGND0260
C                                                                                  LGND0270
      P(1,1,K)=1.0                                                                 LGND0280
C                                                                                  LGND0290
      P(1,2,K)=X                                                                   LGND0300
      P(2,2,K)=SQRT(1.0-XX)                                                        LGND0310
C                                                                                  LGND0320
      P(1,3,K)=0.5*(3.0*XX-1.0)                                                    LGND0330
      P(2,3,K)=3.0*X*P(2,2,K)                                                      LGND0340
      P(3,3,K)=3.0*P(2,2,K)**2                                                     LGND0350
C                                                                                  LGND0360
      IF(MMAX-4)100,10,10                                                          LGND0370
C                                                                                  LGND0380
   10 KOOOFX=MMAX-1                                                                LGND0390
      DO 90 NN=3,KOOOFX                                                            LGND0400
        N=NN+1                                                                     LGND0410
        FN=NN                                                                      LGND0420
        TN=2*NN-1                                                                  LGND0430
      P(1,N,K)=(TN*X*P(1,N-1,K)-(FN-1.0)*P(1,N-2,K))/FN                           LGND0440
C                                                                                  LGND0450
        MAX2=NN-2                                                                  LGND0460
      DO 80 MM=1,MAX2                                                              LGND0470
        M=MM+1                                                                     LGND0480
        SN=NN+MM-1                                                                 LGND0490
        RN=NN-MM                                                                   LGND0500
   80 P(M,N,K)=(TN*X*P(M,N-1,K)-SN*P(M,N-2,K))/RN                                 LGND0510
C                                                                                  LGND0520
        M=N-1                                                                      LGND0530
      P(M,N,K)=TN*X*P(N,M,K)                                                       LGND0540
C                                                                                  LGND0550
```

```
         M1=2*NN+1                                                        LGND0560
         M2=NN+1                                                          LGND0570
      P(N,N,K)=(0.5*P(2,2,K))**NN*FACT(M1,M2)                             LGND0580
C                                                                         LGND0590
   90 CONTINUE                                                            LGND0600
C                                                                         LGND0610
  100 CONTINUE                                                            LGND0620
      RETURN                                                              LGND0630
      END                                                                 LGND0640
$IBFTC CTAU
      SUBROUTINE CTAU
      RETURN
      END
$IBFTC DAUX    REF                                                        DAUX0010
      SUBROUTINE DAUX                                                     DAUX0030
      COMMON    T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
     1 P(11,11,10),PW(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),        DAUX0050
     2 FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),
     3 A(10,10),DELPHI(20),THETA(10)                                      DAUX0070
     4 ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,TAUONE,TAUTWO,
     5  DELTAU,OMEGA,GLBEDO,NEQ,NPHI,FLUX,MPRNT
C                                                                         DAUX0210
      CALL CTAU                                                           DAUX0220
      GO TO (1,2),KFLAG                                                   DAUX0230
    2 DO 22 M=1,MMAX                                                      DAUX0240
         DO 22 I=M,MMAX                                                   DAUX0250
         JO=I-M+1                                                         DAUX0260
         JT=I+M-1                                                         DAUX0270
         J=I+M-2+1
   22 CK(I,M)=C(I)*FACT(JO,JT)*SGN(J)                                     DAUX0290
C                                                                         DAUX0300
    1 CALL ALBEDO                                                         DAUX0310
    3    L=3                                                              DAUX0320
      DO14 M=1,MMAX                                                       DAUX0330
      DO14 K=1,NQUAD                                                      DAUX0340
      DO14 J=1,K                                                          DAUX0350
         L=L+1                                                            DAUX0360
      S(M,K,J)=T(L)                                                       DAUX0370
   14 S(M,J,K)=S(M,K,J)                                                   DAUX0380
      DO 5 M=1,MMAX                                                       DAUX0390
      DO 5 I=M,MMAX                                                       DAUX0400
      DO 5 K=1,NQUAD                                                      DAUX0410
      SUM=0.0                                                             DAUX0420
         DO 6 J=1,NQUAD                                                   DAUX0430
    6    SUM=SUM+S(M,K,J)*PW(M,I,J)                                       DAUX0440
         J=I+M-2+1
    5 PSI(M,I,K)=P(M,I,K)+0.5*SGN(J) *SUM/DEL(M)                          DAUX0460
C                                                                         DAUX0470
      DO 7 M=1,MMAX                                                       DAUX0480
    7 ODEL(M)=OMEGA*DEL(M)                                                DAUX0490
C                                                                         DAUX0500
         J=NEQ+3                                                          DAUX0510
      DO 8 M=1,MMAX                                                       DAUX0520
      DO 8 K=1,NQUAD                                                      DAUX0530
      DO 8 L=1,K                                                          DAUX0540
```

```
      J=J+1                                                             DAUX0550
      SUM=0.0                                                           DAUX0560
        DO 9 I=M,MMAX                                                   DAUX0570
    9   SUM=SUM+CK(I,M)*P-T(M,I,K)*PSI(M,I,L)                           DAUX0580
    8 T(J)=-A(K,L)*S(M,K, .+ODEL(M)*SUM                                 DAUX0590
      RETURN                                                            DAUX0600
      END                                                               DAUX0610
$IBFTC DCTNRY  REF                                                      DCTN0010
      SUBROUTINE DCTNRY                                                 DCTN0030
      COMMON     T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
     1  P(11,11,10),PW(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),     DCTN0050
     2  FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),
     3  A(10,10),DELPHI(20),THETA(10)                                   DCTN0070
     4  ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,NI,TAUONE,TAUTWO,
     5   DELTAU,OMEGA,QLBEDO,NEQ,NPHI,FLUX,MPRNT
C                                                                       DCTN0210
C                     INPUT ROOTS AND WEIGHTS                           DCTN0220
C                                                                       DCTN0230
      DO 1 I=2,10                                                       DCTN0240
    1 READ 100,N,(XL(J,N),J=1,N)                                        DCTN0250
      DO 2 I=2,10                                                       DCTN0260
    2 READ 100,N,(WT(J,N),J=1,N)                                        DCTN0270
  100 FORMAT(I12/(6E12.8))                                              DCTN0280
C                                                                       DCTN0290
C                     SET UP DICTIONARY                                 DC   0300
C                                                                       DCTN0310
C              SINGLE FACTORIALS                                        DCTN0320
      FAC(1)=1.0                                                        DCTN0330
      FAC(2)=1.0                                                        DCTN0340
      FAC(3)=2.0                                                        DCTN0350
      DO 3 J=4,22                                                       DCTN0360
      FJ=J-1                                                            DCTN0370
    3 FAC(J)=FJ*FAC(J-1)                                                DCTN0380
C                                                                       DCTN0390
C              DOUBLE FACTORIALS                                        DCTN0400
      DO 4 J=1,22                                                       DCTN0410
      DO 4 K=1,22                                                       DCTN0420
    4 FACT(J,K)=FAC(J)/FAC(K)                                           DCTN0430
C                                                                       DCTN0440
C              (-1)**(I+M)                                              DCTN0450
      DO 5 M=1,11
      DO 5 I=1,11
      J=M+I-2                                                           DCTN0480
      L=J+1
      MJ=MOD(J,2)+1                                                     DCTN0490
      GO TO (6,7),MJ                                                    DCTN0500
    6 SGN(L)=1.0
      GO TO 5                                                           DCTN0520
    7 SGN(L)=-1.0
    5 CONTINUE                                                          DCTN0540
C                                                                       DCTN0550
C              2.-KRONECKER DELTA(I,M)                                  DCTN0560
      DEL(1)=1.0                                                        DCTN0570
      DO 8 M=2,11                                                       DCTN0580
    8 DEL(M)=2.0                                                        DCTN0590
```

```
C                                                                       DCTN0600
      RETURN                                                            DCTN0610
      END                                                               DCTN0620
$IBFTC SSTART  REF                                                      SSTA0010
      SUBROUTINE SSTART                                                 SSTA0030
      COMMON    T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
     1 P(11,11,10),PW(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),     SSTA0050
     2 FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),
     3 A(10,10),DELPHI(20),THETA(10)                                    SSTA0070
     4 ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,TAUONE,TAUTWO,
     5  DELTAU,OMEGA,QLBEDO,NEQ,NPHI,FLUX,MPRNT
      DO 23 I=1,7263                                                    SSTA0210
   23 T(I)=0.0                                                          SSTA0220
      T(2)=TAUONE                                                       SSTA0230
      T(3)=DELTAU                                                       SSTA0240
      DO 24 I=1,MMAX                                                    SSTA0250
      DO 24 J=1,NQUAD                                                   SSTA0260
      DO 24 K=1,NQUAD                                                   SSTA0270
   24 S(I,J,K)=0.0                                                      SSTA0280
C                                                                       SSTA0290
C                                                                       SSTA0300
      FQ=4.0*QLBEDO                                                     SSTA0310
      DO 26 J=1,NQUAD                                                   SSTA0320
      DO 26 K=1,J                                                       SSTA0330
      S(1,J,K)=FQ*XL(J,NQUAD)*XL(K,NQUAD)                               SSTA0340
   26 S(1,K,J)=S(1,J,K)                                                 SSTA0350
      J=3                                                               SSTA0360
      DO 28 M=1,MMAX                                                    SSTA0370
      DO 28 K=1,NQUAD                                                   SSTA0380
      DO 28 L=1,K                                                       SSTA0390
        J=J+1                                                           SSTA0400
   28 T(J)=S(M,K,L)                                                     SSTA0410
      RETURN                                                            SSTA0420
      END                                                               SSTA0430
$IBFTC OUTPUT  REF                                                      OUTP0010
      SUBROUTINE OUTPUT                                                 OUTP0030
      DIMENSION CMD(20,11)
      COMMON    T(7263),S(11,10,10),Q(11,10,10),ZINT(20,10,10),
     1 P(11,11,10),PW(11,11,10),PSI(11,11,10),XL(10,10),WT(10,10),     ANIS0050
     2 FAC(22),FACT(22,22), SGN(22),DEL(11),ODEL(11),C(11),CK(11,11),
     3 A(10,10),DELPHI(20),THETA(10)
     4 ,MMON,NQUAD,MMAX,NFLAG,KFLAG,LFLAG,NPRNT,N1,TAUONE,TAUTWO,
     5  DELTAU,OMEGA,QLBEDO,NEQ,NPHI,FLUX,MPRNT
C                                                                       OUTP0050
C                                                                       OUTP0060
      IF(LFLAG-1)1,1,6                                                  OUTP0070
C                                                                       OUTP0080
C                STORE ANGLES AND COSINES OF M DELTA PHI                OUTP0090
C                                                                       OUTP0100
    1 QFLUX=0.25*FLUX                                                   OUTP0110
      DO 3 K=1,NQUAD                                                    OUTP0120
        CSTHET=XL(K,NQUAD)                                              OUTP0130
    3 THETA(K)=ARCOS(CSTHET)      *57.2957795
C                                                                       OUTP0160
      DO 5 J=1,NPHI                                                     OUTP0170
```

```
      DELPHI(J)=DELPHI(J)*.174532925E-01
        DO 4 M=1,MMAX
        FM=M-1
        FMD=FM*DELPHI(J)
    4 CMD(J,M)=COS(FMD)
      PRINT1051,J,M,CMD(J,M)
 1051 FORMAT(2I5,E16.8)
    5 CONTINUE
      LFLAG=2
C
C
    6 CALL ALBEDO
      TAU=T(2)
C
C
C            OUTPUT S
C
      PRINT100,OMEGA,QLBEDO,TAU
      DO 10 I=1,NQUAD
      PUNCH200,I
      PRINT103,I
      DO 10 M=1,MMAX
      MM=M-1
      PRINT105,MM,(S(M,J,I),J=1,NQUAD)
  105 FORMAT(3XI2,10F10.6)
   10 PUNCH201,    (S(M,J,I),J=1,NQUAD)
  200 FORMAT(3I12)
  201 FORMAT(6E12.8)
C
C
C
C                   OUTPUT I
C
      PRINT 106,OMEGA,QLBEDO,TAU,(K,K=1,NQUAD)
C
      DO 16 J=1,NPHI
      DO 16 K=1,NQUAD
      DO 16 L=1,K
      SUM=0.0
        DO 14 M=1,MMAX
   14   SUM=SUM+CMD(J,M)*S(M,K,L)
      SUM=SUM*QFLUX
      ZINT(J,L,K)=SUM/XL(L,NQUAD)
   16 ZINT(J,K,L)=SUM/XL(K,NQUAD)
C
      DO 20 L=1,NQUAD
      J=1
      PRINT 103,L, (ZINT(J,K,L),K=1,NQUAD)
      IF(NPHI-1)20,20,19
   19 DO 22 J=2,NPHI
   22 PRINT 108,(ZINT(J,K,L),K=1,NQUAD)
   20 CONTINUE
C
C
  100 FORMAT(1H124X29HSCATTERING COEFFICIENTS, S(M) /
```

```
     1   23X7HOMEGA =,F5.2, 5H, Q =,F5.2, 5H, Z =,F5.2/                      OUTP0550
     2   10X58HFOR THE FOLLOWING POLAR ANGLES OF INCIDENCE AND REFLECTIONOUTP0560
     3   )                                                                    OUTP0570
 101  FORMAT(1H026X5HANGLE,4X7HDEGREES,4X6HCOSINE/(28XI2,F12.2,F11.4))  OUTP0580
 102  FORMAT(1H0/2X8HINCIDENT,20X21HREFLECTED POLAR ANGLE/3X5HANGLE/      OUTP0590
     1   (2X10I10))                                                           OUTP0600
 103  FORMAT(1H02XI2,10F10.6)                                                 OUTP0610
 104  FORMAT(1H03X67HNOTE.   EACH FIGURE ABOVE CORRESPONDS TO AN INCIDENOUTP0620
     1T POLAR ANGLE, A/4X69HREFLECTED POLAR ANGLE, AND A TERM IN THE EXPOUTP0630
     2ANSION OF THE S FUNCTION. //                                            OUTP0640
     34X68HEACH FIGURE ON THE NEXT PAGE CORRESPONDS TO AN INCIDENT POLAROUTP0650
     4 ANGLE,  /4X55HA REFLECTED POLAR ANGLE, AND A CHANGE IN AZIMUTH ANOUTP0660
     5GLE.)                                                                   OUTP0670
 106  FORMAT(1H126X24HSCATTERED INTENSITIES, I /                              OUTP0680
     1   23X7HOMEGA =,F5.2, 5H, Q =,F5.2, 5H, Z =,F5.2 /                      OUTP0690
     2   1H0,1X10I10)                                                         OUTP0700
 108  FORMAT(5X10F10.6)                                                       OUTP0710
      RETURN                                                                  OUTP0720
      END                                                                     OUTP0730
$ENTRY          MAIN
              2                                                                   021
21132486E-0078867514E 00                                                          022
              3                                                                   023
11270166E-0050000000E 0088729834E 00                                              024
              4                                                                   025
69431845E-0133000948E-0066999052E 0093056816E 00                                  026
              5                                                                   027
46910081E-0123076534E-0050000000E 0076923466E 0095308992E 00                      028
              6                                                                   029
33765245E-0116939531E-0038069040E-0061930960E 0083060469E 0096623476E 00          030
              7                                                                   031
25446046E-0112923441E-0029707742E-0050000000E 0070292258E 0087076559E 00          032
97455396E 00                                                                      033
              8                                                                   034
19855071E-0110166676E-0023723390E-0040828268E-0059171732E 0076276620E 00          035
89833324E 0098014493E 00                                                          036
              9                                                                   037
15919883E-0181984445E-0119331428E-0033787329E-0050000000E 0066212671E 00          038
80668572E 0091801555E 0098408012E 00                                              039
             10                                                                   040
13046738E-0167468315E-0116029522E-0028330231E-0042556283E-0057443717E 00          041
71669769E 0083970478E 0093253168E 0098695327E 00                                  042
              2                                                                  0021
50000000E 0050000000E 00                                                         0022
              3                                                                  0023
27777778E-0044444444E-0027777778E-00                                             0024
              4                                                                  0025
17392742E-0032607257E-0032607257E-0017392742E-00                                 0026
              5                                                                  0027
11846343E-0023931433E-0028444444E-0023931433E-0011846343E-00                     0028
              6                                                                  0029
85662244E-0118038078E-0023395696E-0023395696E-0018038078E-0085662244E-01         0030
              7                                                                  0031
64742484E-0113985269E-0019091502E-0020897958E-0019091502E-0013985269E-00         0032
64742484E-01                                                                     0033
```

```
          8                                                                                 0034
50614270E-0111119051E-0015685332E-0018134189E-0018134189E-0015685332E-00                    0035
11119051E-0050614270E-01                                                                    0036
          9                                                                                 0037
40637194E-0190324079E-0113030535E-0015617353E-0016511968E-0015617353E-00                    0038
13030535E-0090324079E-0140637194E-01                                                        0039
         10                                                                                 0040
33335669E-0174725674E-0110954317E-0013463335E-0014776210E-0014776210E-00                    0041
13463335E-0010954317E-0074725674E-0133335669E-01                                            0042
    9    3    1    1    1   20    1    5
         0.0         1.0         .01         1.0         0.0
         1.0         0.0         0.5
        1.        3        0.        90.        180.                                2     4
```

APPENDIX E

PROGRAMS FOR NEUTRON TRANSPORT

<u>PROGRAM E.1. PRODUCTION OF INTERNAL MEASUREMENTS</u>

The complete program is listed:

MAIN program

DAUX subroutine

The following library routine is required:

INTS/INTM

```
$JOB            2609,DYNNEU,K0160,5,100,100,C
$IBJOB          MAP
$IBFTC MAIN     REF
      DIMENSION NPNCH(20),Z(300)
      COMMON T(27),A(20),X(20),IFLAG,AA,U(300),V(300),NSLABS
C
    1 READ(5,100)NPRNT,MPRNT,NSLABS,NGRIDS,NOS
      WRITE(6,90)NPRNT,MPRNT,NSLABS,NGRIDS,NOS
      READ(5,100)(NPNCH(I),I=1,NOS)
      WRITE(6,90)(NPNCH(I),I=1,NOS)
      READ(5,101)DELTA,AA
      WRITE(6,91)DELTA,AA
      READ(5,101)(X(I),I=1,NSLABS)
      WRITE(6,91)(X(I),I=1,NSLABS)
C
C           REFLECTION COEFFICIENT
C
      RC=SIN(AA)/COS(AA)
C
C
C           U AND V FLUXES
C
      T(2)=1.0
      T(3)=-DELTA
      T(4)=RC
      T(5)=1.0
      CALL INTS(T,2,2,0,0,0,0,0,0)
      WRITE(6,94)
      WRITE(6,95)T(2),T(4),T(5)
C
      N=0
      DO 5 I=1,NSLABS
      DO 5 J=1,NGRIDS
      CALL INTM
      WRITE(6,95)T(2),T(4),T(5)
      N=N+1
      U(N)=T(4)
    5 Z(N)=T(2)
C
C           PUNCH U AND V OBSERVATIONS
      PRINT97
      DO  6 M=1,NSLABS
      DO  6 I=1,NOS
      N=(M-1)*NGRIDS+ NPNCH(I)
      PUNCH96,Z(N),U(N)
    6 PRINT96,Z(N),U(N)
C
      GO TO 1
C
  100 FORMAT(6I12)
  101 FORMAT(6E12.8)
   90 FORMAT(1H06I20)
   91 FORMAT(1H06E20.8)
   92 FORMAT(///19X1HX,16X 4HR(X),11X1HA/)
   93 FORMAT(F20.4,E20.8,F12.4)
   94 FORMAT(///19X1HX,16X 4HU(X),16X4HV(X),11X1HA/)
   95 FORMAT(F20.4,2E20.8,F12.4)
   96 FORMA (F12.2,E12.8)
   97 FORMAT(///)
```

```
$IBFTC DAUX     REF
      SUBROUTINE DAUX
      COMMON T(27),A(20),X(20),IFLAG,AA,U(300),V(300),NSLABS
C
    4 T(6)=AA*T(5)
      T(7)=-AA*T(4)
      RETURN
      END
$ENTRY          MAIN
           1        100        10        10         3
           2          5         8
        0.01        0.[illegible]
         0.1        0.2       0.3       0.4       0.5       0.6
         0.7        0.8       0.9       1.0
```

74

PROGRAM E.2. TWO DIMENSIONAL DYNAMIC PROGRAMMING FOR THE DETERMINATION OF ABSORPTION COEFFICIENTS

The complete program is listed:

MAIN program

INTERP subroutine

DAUX subroutine

INTR subroutine

74 The following library routines are required:

BET

INTS/INTM

```
$JOB           2890,DPNT1,K0160,10,100,100,C
$IBJOB         MAP
$IBFTC MAIN    REF
      COMMON T(27),AA,NA,A(51),DA,NC,C(101),DC,NE,E(51),DE,NSLABS,B(51),
     1  NOS,IGRID(100),NGRDSB,MOBS(100),NTOBS,Z(100),W(100),DELTA,MINT,
     2  F(51,51),S(51),H(51,51),U(100)
C
    1 READ(5,100)NA,A(1),DA
      WRITE(6,90)NA,A(1),DA
      DO 2 I=2,NA
    2 A(I)=A(I-1)+DA
C
      READ(5,100)NC,C(1),DC
      WRITE(6,90)NC,C(1),DC
      DO 3 I=2,NC
    3 C(I)=C(I-1)+DC
C
      READ(5,100)NE,E(1),DE
      WRITE(6,90)NE,E(1),DE
      DO 4 I=2,NE
    4 E(I)=E(I-1)+DE
C
      READ(5,100)NSLABS,(B(I),I=1,NSLABS)
      WRITE(6,90)NSLABS,(B(I),I=1,NSLABS)
C
      READ(5,101)NOS,(IGRID(I),I=1,NOS)
      WRITE(6,91)NOS,(IGRID(I),I=1,NOS)
C
      READ(5,101)NGRDSB
      WRITE(6,91)NGRDSB
        N=0
      DO 5 I=1,NSLABS
      DO 5 J=1,NOS
        N=N+1
    5 MOBS(N)=(I-1)*NGRDSB + IGRID(J)
      NTOBS=NOS*NSLABS
      WRITE(6,91)(MOBS(I),I=1,NTOBS)
C
      READ(5,102)(Z(I),W(I),I=1,NTOBS)
      WRITE(6,92)(Z(I),W(I),I=1,NTOBS)
C
      READ(5,102)DELTA
      WRITE(6,92)DELTA
C
      MINT=NGRDSB
C
C           STAGE 1
C
      NSTAGE=1
      WRITE(6,93)NSTAGE
C
      DO 10 I=1,NC
      DO 10 J=1,NE
      AA=ATAN2(C(I),E(J))/B(1)
      F(I,J)=0.0
      DO 6 K=1,NOS
    6 F(I,J)= F(I,J) + (SIN(AA*Z(K)) - W(K))**2
      CP=0.0
      EP=0.0
   10 WRITE(6,94)C(I),E(J),AA,CP,EP,F(I,J)
```

```
C
C
C
      DO 50 NSTAGE=2,NSLABS
      WRITE(6,93)NSTAGE
      DO 40 IC=1,NC
      DO 40 JE=1,NE
C
      DO 30 IA=1,NA
      BX=B(NSTAGE)
      AA=A(IA)
      T(2)=BX
      T(3)=-DELTA
      T(4)=C(IC)
      T(5)=E(JE)
      CALL INTS(T,2,2,0,0,0,0,0,0)
      DO 20 M=1,MINT
      CALL INTM
   20 U(M)=T(4)
      D=0.0
      J=(NSTAGE-1)*NOS
      DO 21 I=1,NOS
      M=IGRID(I)
      J=J+1
   21 D=D + (U(M)-W(J))**2
      CP=T(4)
      EP=T(5)
      CALL INTERP(CP,EP,FI)
   30 S(IA)=D+FI
C
C     MIN S OVER A
      MINA=1
      SMIN=1.0E+20
      DO 31 IA=2,NA
      I=IA
      IF(S(I)-SMIN)32,31,31
   32 SMIN=S(I)
      MINA=I
      AA=A(I)
   31 CONTINUE
      H(IC,JE)=SMIN
C
   40 WRITE(6,94)C(IC),E(JE),AA,CP,EP,H(IC,JE)
      DO 50 IC=1,NC
      DO 50 JE=1,NE
   50 F(IC,JE)=H(IC,JE)
C
      GO TO 1
C
  100 FORMAT(I12,5E12.8/(6E12.8))
   90 FORMAT(1H0I20,5E20.8/(6E20.8))
  101 FORMAT(6I12)
   91 FORMAT(1H06I20)
  102 FORMAT(2E12.8)
   92 FORMAT(1H06E20.8)
   93 FORMAT(1H1 9HSTAGE N =, I3//18X2HC1,18X2HC2,19X1HA,17X3HC1',
     1        17X3HC2',12X8HF(C1,C2) )
   94 FORMAT(6E20.8)
      END
$IBFTC INTERP  REF
```

```
      SUBROUTINE INTERP(X,Y,ANS)
      COMMON T(27),AA,NA,A(51),DA,NC,C(101),DC,NE,E(51),DE,NSLABS,B(51),
     1  NOS,IGRID(100),NGRDSB,MOBS(100),NTOBS,Z(100),W(100),DELTA,MINT,
     2  F(51,51),S(51),H(51,51)
C            TWO-DIM. INTERPOLATION
C     FIND I1,I2, I.E., X1, X2
      DO 1 I=2,NC
      I1=I
      I2=I-1
      X1=C(I1)
      X2=C(I2)
      IF(BET(X1,X,X2,MM))1,2,2
    1 CONTINUE
      ANS=1.0E+20
      RETURN
C
C     FIND J1,J2, I.E., Y1, Y2
    2 DO 11 J=2,NE
      J1=J
      J2=J-1
      Y1=E(J1)
      Y2=E(J2)
      IF(BET(Y1,Y,Y2,MM))11,12,12
   11 CONTINUE
      ANS=1.0E+20
      RETURN
C
C     FIND F(X,Y1)=G1
   12 F1=F(I1,J1)
      F2=F(I2,J1)
      DX=X2-X1
      D=X-X1
      CALL INTR(F1,F2,DX,D,G1)
C
C     FIND F(X,Y2)=G2
      F1=F(I1,J2)
      F2=F(I2,J2)
      CALL INTR(F1,F2,DX,D,G2)
C
C     FIND F(X,Y)=ANS
      DY=Y2-Y1
      D=Y-Y1
      CALL INTR(G1,G2,DY,D,ANS)
      RETURN
      END
$IBFTC DAUX    REF
      SUBROUTINE DAUX
      COMMON T(27),AA,NA,A(51),DA,NC,C(101),DC,NE,E(51),DE,NSLABS,B(51),
     1  NOS,IGRID(100),NGRDSB,MOBS(100),NTOBS,Z(100),W(100),DELTA,MINT,
     2  F(51,51),S(51),H(51,51)
C
      T(6)=AA*T(5)
      T(7)=-AA*T(4)
      RETURN
      END
$IBFTC INTR    REF
      SUBROUTINE INTR(F1,F2,DX,D,G)
C            ONE-DIM. INTERPOLATION
      G=F1 + (F2-F1)*D/DX
      RETURN
```

```
END
     10        0.1        0.1
     21        0.0        0.1
     21        1.0       0.05
     10        0.1        0.2        0.3        0.4        0.5
    0.6        0.7        0.8        0.9        1.0
      3          2          5          8
     10
  0.0211394757E-01
  0.0528484388E-01
  0.0845567610E-01
  0.1268328626E-01
  0.1585381951E-01
  0.1810241607E-00
  0.2212509171E-00
  0.2514206610E-00
  0.2815900853E-00
  0.3218154213E-00
  0.3519839517E-00
  0.3821520356E-00
  0.4223753879E-00
  0.4525422834E-00
  0.4827086070E-00
  0.5229294173E-00
  0.5530942608E-00
  0.5832584082E-00
  0.6234761246E-00
  0.6536385042E-00
  0.6838000651E-00
  0.7240141434E-00
  0.7541736532E-00
  0.7843322238E-00
  0.8245421290E-00
  0.8546983702E-00
  0.8848535543E-00
  0.9250587616E 00
  0.9552113437E 00
  0.9853627533E 00
  0.01
```

PROGRAM E.3. ONE DIMENSIONAL DYNAMIC PROGRAMMING FOR THE DETERMINATION OF ABSORPTION COEFFICIENTS

The complete program is listed:

MAIN program

DAUX subroutine

SUBREF subroutine

SHIFT subroutine

SUBNLV subroutine

SUBDF subroutine

The following library routines are required:

BET

INTS/INTM

```
$IBFTC MAIN     REF
      COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
     1  IGRID(50),NGRDSB,DELTA,MOBS(200),NTOBS,Z(200),W(200),MINT,
     2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,TBIG,
     3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
C            INPUT
C
    1 READ(5,100)NA,A(1),DA
      WRITE(6,90)NA,A(1),DA
      DO 2 I=2,NA
    2 A(I)=A(I-1)+DA
C
      READ(5,100)NC,C(1),DC
      WRITE(6,90)NC,C(1),DC
      DO 3 I=2,NC
    3 C(I)=C(I-1)+DC
C
      READ(5,100)NSLABS,B(1),DB
      WRITE(6,90)NSLABS,B(1),DB
      DO 4 I=2,NSLABS
    4 B(I)=B(I-1)+DB
C
      READ(5,101)NOS,(IGRID(I),I=1,NOS)
      WRITE(6,91)NOS,(IGRID(I),I=1,NOS)
C
      READ(5,100)NGRDSB,DELTA,ALPHA
      WRITE(6,90)NGRDSB,DELTA,ALPHA
        N=0
      DO 5 I=1,NSLABS
      DO 5 J=1,NOS
        N=N+1
    5 MOBS(N)=(I-1)*NGRDSB + IGRID(J)
      NTOBS=NOS*NSLABS
      WRITE(6,91)(MOBS(N),N=1,NTOBS)
C
      READ(5,102)(Z(I),W(I),I=1,NTOBS)
      WRITE(6,92)(Z(I),W(I,,I=1,NTOBS)
C
      MINT=NGRDSB
C            STAGE 1
      NSTAGE=1
      WRITE(6,93) NSTAGE
      DO 8  IC=1,NC
      SMIN=1.0E+10
      DO 7  IA=1,NA
C
C            FIND CP=V(0)
C
      IFLAG=1
      AA=A(IA)
      T(4)=0.0
      T(5)=1.0
      T(2)=0.0
      T(3)=DELTA
      CALL INTS(T,2,2,0,0,0,0,0,0)
      DO 6 M=1,MINT
    6 CALL INTM
      CP=C(IC)/T(5)
      EP=0.0
```

```
   10 CALL INTM
      EP=(C(IC)-CP*T(7)) / T(5)
C
C              COMPUTE D,F
      CALL SUBDF
   11 CONTINUE
      AA=AMIN
      F(IC)=SMIN
      IF(F(IC)-100.0)16,15,15
   15 NC=IC-1
      WRITE(6,95)NC
      GO TO 17
C
C              COMPUTE R(C)
   16 CALL SUBREF
      R(IC)=RP
   12 WRITE(6,94)C(IC),AA,CPA,EPA,R(IC),F(IC)
   17 CALL SHIFT
   13 CONTINUE
      GO TO 1
C
  100 FORMAT(I12,5E12.8/(6E12.8))
   90 FORMAT(1H0I20,5E20.8/(6E20.8))
  101 FORMAT(6I12)
   91 FORMAT(1H06I20)
  102 FORMAT(2E12.8)
   92 FORMAT(1H06E20.8)
   93 FORMAT(1H19HSTAGE N =, I3//19X1HC,19X1HA,18X2HCP,18X2HEP,
     1          16X4HR(C),16X4HF(C)//)
   94 FORMAT(2F20.6,4E20.6)
   95 FORMAT(1X18HNUMBER OF STATES =, I5)
      END
$IBFTC DAUX     REF
      SUBROUTINE DAUX
      COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
     1  IGRID(50),NGRDSB,DELTA,MOBS(200),NTOBS,Z(200),W(200),MINT,
     2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,IBIG,
     3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
      GO TO(1,2,3,4),IFLAG
C
C     TRANSPORT EQS. FOR U, V
C
    1 T(6)= AA*T(5)
      T(7)=-AA*T(4)
      RETURN
C
C              FOR P, H
    2 T(8) = AA*T(5)
      T(9) =-AA*T(4)
      T(10)= AA*T(7)
      T(11)=-AA*T(6)
      RETURN
C
C     REFLECTION
C
    3 T(5)=AA*(1.0 + T(4)**2)
      RETURN
C
C     AND TRANSMISSION
```

```
    4  T(6)=AA*(1.0 + T(4)**2)
       T(7)=AA*T(4)*T(5)
       RETURN
       END
$IBFTC SUBREF
       SUBROUTINE SUBREF
       COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
      1  IGRID(50),NGRDSB,DELTA,MOBS(200),NIOBS,Z(200),W(200),MINI,
      2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,TBIG,
      3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
C            COMPUTE R(N)
C
       IFLAG=3
       T(2)=0.0
       T(3)=DELTA
       T(4)=RP
       CALL INTS(T,1,2,0,0,0,0,0,0)
       DO 1 M=1,MINT
    1  CALL INTM
       RP=T(4)
       RETURN
       END
$IBFTC SHIFT   REF
       SUBROUTINE SHIFT
       COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
      1  IGRID(50),NGRDSB,DELTA,MOBS(200),NIOBS,Z(200),W(200),MINI,
      2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,TBIG,
      3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
       DO 1 I=1,NC
       RO(I)=R(I)
    1  FO(I)=F(I)
       RETURN
       END
$IBFTC SUBNLV  REF
       SUBROUTINE SUBNLV(CC,IS)
       COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
      1  IGRID(50),NGRDSB,DELTA,MOBS(200),NIOBS,Z(200),W(200),MINI,
      2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,TBIG,
      3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
C            SOLVE N.L. B.C. FOR  CP
       IS=1
       J=1
       Z1=RBIG*C(J)*RO(J)
       Z2=C(J)-CC*TBIG
       DIF1=Z1-Z2
       IF(DIF1)1,2,3
    2  CP=C(J)
       RP=RO(J)
       RETURN
C            DIF1 IS NEG.
    1  DO 11 J=2,NC
       J2=J
       J1=J-1
       Z1=RBIG*C(J)*RO(J)
       Z2=C(J)-CC*TBIG
       DIF2=Z1-Z2
       IF(DIF2)10,2,12
```

```
   10 DIF1=DIF2
   11 CONTINUE
      GO TO 13
   12 CP=C(J1) +  DIF1*DC/(DIF1-DIF2)
      RP=RO(J1) + (CP-C(J1))*(RO(J2)-RO(J1))/DC
      RETURN
C              DIF1 IS POS.
    3 DIF1=-DIF1
      DO 21 J=2,NC
      J2=J
      J1=J-1
      Z1=RBIG*C(J)*RO(J)
      Z2=C(J)-CC*TBIG
      DIF2=Z2-Z1
      IF(DIF2)20,2,12
   20 DIF1=DIF2
   21 CONTINUE
   13 IS=0
      RETURN
      END
$IBFTC SUBDF   REF
      SUBROUTINE SUBDF
      DIMENSION U(200)
      COMMON T(51),NA,A(10),DA,NC,C(1001),DC,NSLABS,B(10),DB,NOS,
     1  IGRID(50),NGRDSB,DELTA,MOBS(200),NTOBS,Z(200),W(200),MINT,
     2  IFLAG,AA,CP,EP,CPA,EPA,R(1001),F(1001),NSTAGE,RBIG,TBIG,
     3  RP,RO(1001),ALPHA,SMIN,AMIN,FO(1001)
C
C              INTEGRATE TRANSPORT EQS.
C              COMPUTE D AND CURRENT F
C
      IFLAG=1
      T(2)=0.0
      T(3)=DELTA
      T(4)=EP
      T(5)=CP
      CALL INTS(T,2,2,0,0,0,0,0,0)
      DO 1 M=1,MINT
      CALL INTM
    1 U(M)=T(4)
C
      D=0.0
      J=(NSTAGE-1)*NOS
      DO 2 I=1,NOS
      M=IGRID(I)
      J=J+1
    2 D=D + (U(M)-W(J))**2
C
      IF(NSTAGE-1)3,3,6
C
    3 S=D*ALPHA
   10 IF(S-SMIN)4,5,5
    4 SMIN=S
      AMIN=AA
      CPA=CP
      EPA=EP
    5 RETURN
C              INTERPOLATE FOR F(N-1)
    6 DO 7 I=2,NC
      I1=I-1
```

```
C
C           COMPUTE D, F
C
    7  CALL SUBDF
       AA=AMIN
       F(IC)=SMIN
C
C           COMPUTE R(C)
C
       RP=0.0
       CALL SUBREF
       R(IC)=RP
       WRITE(6,94)C(IC),AA,CPA,EPA,R(IC),F(IC)
    8  CALL SHIFT
C
C           GENERAL STAGE
C
       DO 13  N=2,NSLABS
       NSTAGE=N
       WRITE(6,93)NSTAGE
       DO 12  IC=1,NC
       SMIN=1.0E+20
       AMIN=0.0
       CPA=0.0
       EPA=0.0
       R(IC)=0.0
       F(IC)=0.0
C
       DO 11  IA=1,NA
C
C           FIND RBIG,TBIG
       AA=A(IA)
       IFLAG=4
       T(2)=0.0
       T(3)=DELTA
       T(4)=0.0
       T(5)=1.0
       CALL INTS(T,2,2,0,0,0,0,0,0)
       DO 9 M=1,MINT
    9  CALL INTM
       RBIG=T(4)
       TBIG=T(5)
C
C           FIND CP=V(N-1), RP=R(V)
C
       CC=C(IC)
       CALL SUBNLV(CC,IS)
       IF(IS)11,11,14
C
C           FIND EP=U(N-1)
C
   14  IFLAG=2
       T(2)=0.0
       T(3)=DELTA
       T(4)=1.0
       T(5)=0.0
       T(6)=0.0
       T(7)=1.0
       CALL INTS(T,4,2,0,0,0,0,0,0)
       DO 10 M=1,MINT
```

```
      I2=I
      X1=C(I1)
      X2=C(I2)
      IF(BET(X1,CP  ,X2,MM))7,8,8
    7 CONTINUE
      S=1.0E+10
      GO TO 10
C
    8 F1=FO(I1)
      F2=FO(I2)
      DX=X2-X1
      G=CP  -X1
      FX=F1  + (F2-F1)*G/DX
      S=D*FX
      GO TO 10
      END
```

APPENDIX F

PROGRAMS FOR WAVE PROPAGATION:

6

MEASUREMENTS OF TRANSIENTS

PROGRAM F.1. DETERMINATION OF WAVE VELOCITY FOR EXAMPLE 1 – HOMOGENEOUS MEDIUM. STEP FUNCTION FORCE

The complete program is listed:

MAIN program

LAPLAC subroutine

DAUX subroutine

INITL subroutine

PANDH subroutine

LINEAR subroutine

NEXT subroutine

OUTPUT subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$IBFTC MAIN     LIST
C
C             VIBRATING STRING - LAPLACE TRANSFORMS
C
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4  ,MX,FORCTX(9),ATRUE,CSPEED
C
C             INPUT
C
    1 READ100,NT,NPRNT,MPRNT,KMAX
      PRINT90,NT,NPRNT,MPRNT,KMAX
      READ101,(RT(I),I=1,NT)
      PRINT91,(RT(I),I=1,NT)
      READ101,(WT(I),I=1,NT)
      PRINT92,(WT(I),I=1,NT)
      READ101,DELTA,TENSN,A,ATRUE
      PRINT93,DELTA,TENSN,A,ATRUE
      READ101,(UOBS1(I),I=1,NT)
      PRINT94,(UOBS1(I),I=1,NT)
      READ101,(FORCE(I),I=1,NT)
      PRINT95,(FORCE(I),I=1,NT)
      NP1=NT+1
      NTWO=2*NT
      DO 11 I=1,NT
   11 U(I,1)=0.0
      READ101,(U(I,1),I=NP1,NTWO)
      PRINT96,(U(I,1),I=NP1,NTWO)
C
C
C             PRODUCE TRANSFORS OF OBSERVATIONS
C
    2 CALL LAPLAC
C
C             GENERATE INITIAL APPROXIMATION
C
    3 CALL INITL
C
C             SUCCESSIVE APPROXIMATIONS
C
    4 DO 5 K=1,KMAX
      PRINT97,K
      CALL PANDH
      CALL LINEAR
    5 CALL NEXT
      GO TO 1
C
  100 FORMAT(6I12)
  101 FORMAT(6E12.8)
   90 FORMAT(1H14X 4HNT =,I3/5X 7HNPRNT =,I3/5X 7HMPRNT =,I3/
     1        5X 6HKMAX =,I3)
   91 FORMAT(1H04X 5HROOTS/(5X6E20.8))
   92 FORMAT(1H04X 7HWEIGHTS/(5X6E20.8))
```

```
   93 FORMAT(1H04X 7HDELTA =,E16.8,5X 7HTENSN =,E16.8,
     1        5X20HINITIAL GUESS OF A =,E16.8/ 5X8HTRUE A =, E16.8)
   94 FORMAT(1H04X12HOBSERVATIONS/(5X6E20.8))
   95 FORMAT(1H04X12HFORCE F(T)  /(5X6E20.8))
   96 FORMAT(1H04X25HINITIAL GUESS OF U-PRIMED/(5X6E20.8))
   97 FORMAT(1H14X13HAPPROXIMATION, I3//)
      END
$IBFTC LAPLAC  LIST
      SUBROUTINE LAPLAC
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1 ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2 ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4 ,MX,FORCTX(9),ATRUE,CSPEED
C
C           THE TIMES
C
      DO 1 I=1,NT
    1 TT(I)=-ALOG(RT(I))
C           THE TRANSFORMS
      DO 2 IS=1,NT
      UOBST(IS)=0.0
      FORCT(IS)=0.0
      JJ=IS-1
      DO 2 I=1,NT
      RW=WT(I)*(RT(I)**JJ)
      UOBST(IS)= UOBST(IS) + UOBS1(I)*RW
    2 FORCT(IS)= FORCT(IS) + FORCE(I)*RW
C
      PRINT10
   10 FORMAT(///1H013X1HT,11X 9HUOBS(1,T),16X 4HF(T),
     1        14X 1HS, 6X14HUOBSTRANS(1,S), 16X 4HF(S) /)
      DO 3 I=1,NT
    3 PRINT11, TT(I), UOBS1(I), FORCE(I), I, UOBST(I), FORCT(I)
   11 FORMAT( 5XF10.6, 2E20.8,10X,I5,2E20.8)
C
C          EXACT TRANSFORMS OF OBSERVATIONS
      CSPEED=SQRT(ATRUE)
      COVERT=CSPEED/TENSN
      DO 6 IS=1,NT
      S=IS
      FORCTX(IS)=1.0/S
    6 UOBSTX(IS)=TANH(S/CSPEED)*FORCTX(IS)*COVERT/S
      PRINT98, (UOBSTX(IS),IS=1,NT)
   98 FORMAT(1H04X32HEXACT TRANSFORMS OF OBSERVATIONS/(5X6E20.8))
      PRINT99, (FORCTX(IS),IS=1,NT)
   99 FORMAT(1H04X25HEXACT TRANSFORMS OF FORCE / (5X6E20.8))
      RETURN
      END
$IBFTC DAUX    LIST
      SUBROUTINE DAUX
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1 ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2 ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
```

```
      4   ,MX,FORCTX(9),ATRUE,CSPEED
       DIMENSION V(19)
       GO TO (100,200,200),LFLAG
C
C             NONLINEAR
C
 100      L=3
       DO 1 IS=1,N21
          L=L+1
    1  V(IS)=T(L)
          L=NEQ+3
       DO 2 IS=1,NT
          L=L+1
       NN=NT+IS
    2  T(L)=V(NN)
       DO 3 IS=1,NT
          L=L+1
          S=IS**2
    3  T(L)=S*V(IS)/T(NEQ+3)
          L=L+1
       T(L)=0.0
       RETURN
C
C             LINEAR
C
 200      L=3
       DO 4 IS=1,N21
          L=L+1
    4  V(IS)=T(L)
          M=NEQ+3
       DO 5 IS=1,NT
          M=M+1
       NN=NT+IS
    5  T(M)=V(NN)
       DO 6 IS=1,NT
          M=M+1
          S=IS**2
    6  T(M)=S*(V(IS) -  V(N21)*UPREV(IS)/A + UPREV(IS))/A
          M=M+1
       T(M)=0.0
C
       IF(LFLAG-3) 20,300,300
 300   RETURN
C
C             HOMOGENEOUS
C
  20   DO 0 J=1,NP1
C
       DO 7 IS=1,N21
          L=L+1
    7  V(IS)=T(L)
C
       DO 8 IS=1,NT
          M=M+1
       NN=NT+IS
```

```
    8   T(M)=V(NN)
 C
        DO 9 IS=1,NT
          M=M+1
          S=IS**2
    9   T(M)=S*(V(IS) - V(N21)*UPREV(IS)/A )/A
          M=M+1
   10   T(M)=0.0
        RETURN
        END
$IBFTC INITL   LIST
        SUBROUTINE INITL
        COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
       1   ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
       2   ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
       3   ,TT(9),U(18,401),P(19),H(19,10),C(10)
       4   ,MX,FORCTX(9),ATRUE,CSPEED
C
C              INITIAL APPROXIMATION FROM NONLINEAR EQUATIONS
C
        LFLAG=1
        DO 1 I=1,2511
    1   T(I)=0.0
        T(3)=DELTA
          L=NT+3
        DO 2IS=1,NT
          J=NT + IS
          L=L+1
    2   T(L)=U(J,1)
          L=L+1
        T(L)=A
C
        I=1
        N21=2*NT + 1
        NEQ=N21
        CALL INTS(T,NEQ,2,0,0,0,0,0,0)
        MX=I
        CALL OUTPUT
C
        DO 4 M1=1,MPRNT
        DO 3 M2=1,NPRNT
        CALL INTM
        I=I+1
          L=3
        DO 3 IS=1,NTWO
          L=L+1
    3   U(IS,I)=T(L)
        MX=I
    4   CALL OUTPUT
        RETURN
        END
$IBFTC PANDH   LIST
        SUBROUTINE PANDH
        COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
       1   ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
```

```
     2   ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3   ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4   ,MX,FORCTX(9),ATRUE,CSPEED
C
      LFLAG=2
      DO 1 I=1,2511
    1 T(I)=0.0
      T(3)=DELTA
      NEQ=(NT+2)*N21
        L=4 + N21 + NT
      T(L)=1.0
      DO 2 I=1,NT
        L=L + N21 + 1
    2 T(L)=1.0
        I=1
      DO 12 IS=1,NT
   12 UPREV(IS)=U(IS,I)
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
        NEQ3=NEQ+3
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
      CALL INTM
        I=I+1
      DO  3 IS=1,NT
    3 UPREV(IS)=U(IS,I)
    4 CONTINUE
C
        L=3
      DO 5 IS=1,N21
        L=L+1
    5 P(IS)=T(L)
C
      DO 6 J=1,NP1
      DO 6 IS=1,N21
        L=L+1
    6 H(IS,J)=T(L)
C
   10 FORMAT(F20.6,5E20.8/(5E20.8))
      RETURN
      END
$IBFTC LINEAR  LIST
      SUBROUTINE LINEAR
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1   ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2   ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3   ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4   ,MX,FORCTX(9),ATRUE,CSPEED
      DIMENSION AM(50,50),BV(50),IPIVOT(50),PIVOT(50),INDEX(50,2)
C
      DO 2 I=1,NT
        II=NT+I
      DO 1 J=1,NP1
    1 AM(I,J)=H(II,J)
    2 BV(I)=FORCT(I)/TENSN - P(II)
C
```

```
C
        I=NT+1
      DO 3 J=1,NP1
      AM(I,J)=0.0
      DO 3 IS=1,NT
    3 AM(I,J)=AM(I,J) + H(IS,J)*H(IS,NP1)
      BV(I)=0.0
      DO 4 IS=1,NT
    4 BV(I)=BV(I) + (UOBST(IS) - P(IS))*H(IS,NP1)
C
      DO 5 I=1,NP1
    5 PRINT9,(AM(I,J),J=1,NP1),BV(I)
    9 FORMAT(5X6E20.8)
C
      CALL MATINV(AM,NP1,BV,1,DETERM,PIVOT,INDEX,IPIVOT)
C
      DO 6 I=1,NP1
    6 C(I)=BV(I)
      A=C(NP1)
      PRINT9,(C(I),I=1,NP1)
      RETURN
      END
$IBFTC NEXT    LIST
      SUBROUTINE NEXT
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1 ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2 ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4 ,MX,FORCTX(9),ATRUE,CSPEED
C
      LFLAG=3
      NEQ=N21
      DO 1 I=1,2511
    1 T(I)=0.0
      T(3)=DELTA
        L=NT+3
      DO 2 I=1,NP1
      J=NT+I
      U(J,1)=C(I)
        L=L+1
    2 T(L)=C(I)
        I=1
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      MX=1
      CALL OUTPUT
C
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
      CALL INTM
        I=I+1
        L=3
      DO 3 IS=1,NTWO
        L=L+1
    3 U(IS,I)=T(L)
      MX=I
```

```
    4 CALL OUTPUT
      RETURN
      END
$IBFTC OUTPUT  LIST
      SUBROUTINE OUTPUT
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEG
     2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4  ,MX,FORCTX(9),ATRUE,CSPEED
C
      I=MX
      PRINT10,T(2)
   10 FORMAT(1H04X 3HX =, F10.6)
      PRINT11,(U(IS,I),IS=1,NT)
      PRINT11,(U(IS,I),IS=NP1,NTWO)
   11 FORMAT(1H04X6E20.8/(5X6E20.8))
      RETURN
      END
```

PROGRAM F.2. DETERMINATION OF WAVE VELOCITY FOR EXAMPLE 2 - HOMOGENEOUS MEDIUM. DELTA-FUNCTION FORCE

A partial program is listed:

LAPLAC subroutine

The following subroutines are required from Program F.1:

MAIN program

DAUX subroutine

INITL subroutine

PANDH subroutine

LINEAR subroutine

NEXT subroutine

OUTPUT subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$IBFTC LAPLAC  LIST
      SUBROUTINE LAPLAC
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N21,NEQ
     2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4  ,MX,FORCTX(9),ATRUE,CSPEED
C
C              FOR DELTA FUNCTION FORCE
C
C
C              THE TIMES
C
      DO 1 I=1,NT
    1 TT(I)=-ALOG(RT(I))
C
C              EXACT TRANSFORMS OF OBSERVATIONS
      CSPEED=SQRT(ATRUE)
      COVERT=CSPEED/TENSN
      DO 6 IS=1,NT
      S=IS
      FORCTX(IS)=1.0
    6 UOBSTX(IS)=TANH(S/CSPEED)*FORCTX(IS)*COVERT/S
      PRINT98, (UOBSTX(IS),IS=1,NT)
   98 FORMAT(1H04X32HEXACT TRANSFORMS OF OBSERVATIONS/(5X6E20.8))
      PRINT99, (FORCTX(IS),IS=1,NT)
   99 FORMAT(1H04X25HEXACT TRANSFORMS OF FORCE / (5X6E20.8))
C
      DO 200 I=1,NT
  200 FORCT(I)=FORCTX(I)
C
C              THE TRANSFORMS
      DO 2 IS=1,NT
      UOBST(IS)=0.0
      JJ=IS-1
      DO 2 I=1,NT
      RW=WT(I)*(RT(I)**JJ)
    2 UOBST(IS)= UOBST(IS) + UOBS1(I)*RW
C
      PRINT10
   10 FORMAT(///1H013X1HT,11X 9HUOBS(1,T),16X 4HF(T),
     1        14X 1HS, 6X14HUOBSTRANS(1,S), 16X 4HF(S) /)
      DO 3 I=1,NT
    3 PRINT11, TT(I), UOBS1(I), FORCE(I), I, UOBST(I), FORCT(I)
   11 FORMAT( 5XF10.6, 2E20.8,10X,I5,2E20.8)
      RETURN
      END
```

PROGRAM F.3. PRODUCTION OF OBSERVATIONS FOR EXAMPLE 3 – INHOMOGENEOUS MEDIUM WITH DELTA FUNCTION INPUT

The complete program is listed:

MAIN program

DAUX subroutine

The following library routines are required:

MATINV

INTS/INTM

```
$JOB            26.9,VIBR3,K0160,5,100,100,C
$IBJOB          MAP
$IBFTC MAIN     REF
C
C PHASE I.  INHOMOGENEOUS STRING, LINEAR PROFILE. LAPLACE TRANSFORMS.
C
      DIMENSION C(50,50),D(50),'PIVOT(50),PIVOT(50),INDEX(50,2)
      COMMON T(1945),H(18,9),     N2,NT,NPRNT,MPRNT,DELTA,A,B,U(18)
C
C               INPUT
C
    1 READ100,NT,NPRNT,MPRNT
      PRINT90,NT,NPRNT,MPRNT
      READ101,DELTA,A,B
      PRINT91,DELTA,A,B
      N2=2*NT
      NEQ=N2*NT
C
C               INITIALIZE
C
      DO 2 I=1,1945
    2 T(I)=0.0
      T(3)=DELTA
      DO 3 J=1,NT
      DO 4 I=1,N2
    4 H(I,J)=0.0
        K=NT + J
    3 H(K,J)=1.0
        L=3
      DO 5 J=1,NT
      DO 5 I=1,N2
        L=L+1
    5 T(L)=H(I,J)
C
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      N3=NEQ+3
      PRINT92,T(2),(T(I),I=4,N3)
C
C               INTEGRATE
C
      DO 10 M1=1,MPRNT
      DO  9 M2=1,NPRNT
    9 CALL INTM
   10 PRINT93,T(2),(T(I),I=4,N3)
C
C               LINEAR SYSTEM
C
        L=3
      DO 11 J=1,NT
      DO 11 I=1,N2
        L=L+1
   11 H(I,J)=T(L)
      DO 12 I=1,NT
      II=NT+I
      DO 12 J=1,NT
   12 C(I,J)=H(II,J)
      DO 13 I=1,NT
   13 D(I)=1.0
      DO 15 I=1,NT
   15 PRINT97,(C(I,J),J=1,NT)
```

```
C
      CALL MATINV(C,NT,D,1,DETERM,PIVOT,INDEX,IPIVOT)
      PRINT94,(I,D(I) I=1,NT)
C
C             OBSERVATIONS   UOBST(1,S)
C
      DO 14 I=1,NT
      U(I)=0.0
      DO 14 J=1,NT
   14 U(I)=U(I) + D(J)*H(I,J)
      PRINT95,(I,U(I),I=1,NT)
      PUNCH96,(U(I),I=1,NT)
      GO TO 1
C
C
  100 FORMAT(6I12)
  101 FORMAT(6E12.8)
   90 FORMAT(1H14X,6I20)
   91 FORMAT(1H04X,6E20.8)
   92 FORMAT(///5X3HX =,F7.3/(2X7E18.8))
   93 FORMAT(1H04X3HX =,F7.3/(2X7E18.8))
   94 FORMAT(///19X1HI,22X8HSLOPE(I)/(15X15,E30.8))
   95 FORMAT(///19X1HI,22X8HUOBST(I)/(15X15,E30.8))
   96 FORMAT(6E12.6)
   97 FORMAT(2X7E18.8)
      END
$IBFTC DAUX     REF
      SUBROUTINE DAUX
      COMMON T(1945),H(18,9),      N2,NT,NPRNT,MPRNT,DELTA,A,B,U(18)
C
        L=3
      DO 1 J=1,NT
      DO 1 I=1,N2
        L=L+1
    1 H(I,J)=T(L)
      DENOM=A + B*T(2)
      DO 3 J=1,NT
      DO 2 I=1,NT
        L=L+1
        II=NT+I
    2 T(L)=H(II,J)
      DO 3 I=1,NT
        L=L+1
        F=I**2
    3 T(L)=F*H(I,J)/DENOM
      RETURN
      END
```

PROGRAM F.4. DETERMINATION OF WAVE VELOCITY FOR EXAMPLE 3 — INHOMOGENEOUS MEDIUM WITH DELTA FUNCTION INPUT

The complete program is listed:

MAIN program

INITL subroutine

PANDH subroutine

LINEAR subroutine

OUTPUT subroutine

DAUX subroutine

NEXT subroutine

The folliwng library routines are required:

MATINV

INTS/INTM

```
$IBFTC MAIN    REF
C
C           VIBRATING STRING - LAPLACE TRANSFORMS - C**2 = A + B*X
C
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1 ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
     2 ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4 ,MX,FORCTX(9),ATRUE,CSPEED
C
C           INPUT
C
    1 READ100,NT,NPRNT,MPRNT,KMAX
      PRINT90,NT,NPRNT,MPRNT,KMAX
      READ101,(RT(I),I=1,NT)
      PRINT91,(RT(I),I=1,NT)
      READ101,(WT(I),I=1,NT)
      PRINT92,(WT(I),I=1,NT)
      READ101,DELTA,TENSN,A,ATRUE,B,BTRUE
      PRINT93,DELTA,TENSN,A,ATRUE,B,BTRUE
      READ101,(UOBST(I),I=1,NT)
      PRINT94,(UOBST(I),I=1,NT)
      READ101,(FORCT(I),I=1,NT)
      PRINT95,(FORCT(I),I=1,NT)
      NTWO=2*NT
      N22=NTWO + 2
      NP1=NT+1
      NP2=NT+2
      DO 11 I=1,NT
   11 U(I,1)=0.0
      READ101,(U(I,1),I=NP1,NTWO)
      PRINT96,(U(I,1),I=NP1,NTWO)
C
C
C           GENERATE INITIAL APPROXIMATION
C
    3 CALL INITL
C
C           SUCCESSIVE APPROXIMATIONS
C
    4 DO 5 K=1,KMAX
      PRINT97,K
      CALL PANDH
      CALL LINEAR
    5 CALL NEXT
      GO TO 1
C
  100 FORMAT(6I12)
  101 FORMAT(6E12.8)
   90 FORMAT(1H14X 4HNT =,I3/5X 7HNPRNT =,I3/5X 7HMPRNT =,I3/
     1        5X 6HKMAX =,I3)
   91 FORMAT(1H04X 5HROOTS/(5X6E20.8))
   92 FORMAT(1H04X 7HWEIGHTS/(5X6E20.8))
   93 FORMAT(1H04X 7HDELTA =,E16.8,5X 7HTENSN =,E16.8,
     1        5X20HINITIAL GUESS OF A =,E16.8, 5X8HTRUE A =, E16.8/
     2        5X20HINITIAL GUESS OF B =,E16.8, 5X8HTRUE B =, E16.8)
   94 FORMAT(1H04X12HOBSERVATIONS/(5X6E20.8))
   95 FORMAT(1H04X12HFORCE F(T)   /(5X6E20.8))
   96 FORMAT(1H04X25HINITIAL GUESS OF U-PRIMED/(5X6E20.8))
   97 FORMAT(1H14X13HAPPROXIMATION, I3//)
```

```
      END
$IBFTC INITL    LIST
      SUBROUTINE INITL
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1   ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
     2   ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3   ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4   ,MX,FORCTX(9),ATRUE,CSPEED
C
C             INITIAL APPROXIMATION FROM NONLINEAR EQUATIONS
C
      LFLAG=1
      DO 1 I=1,2511
    1 T(I)=0.0
      T(3)=DELTA
        L=NT+3
      DO 2IS=1,NT
        J=NT + IS
        L=L+1
    2 T(L)=U(J,1)
        L=L+1
      T(L)=A
        L=L+1
      T(L)=B
C
      I=1
      N22=2*NT + 2
      NEQ=N22
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      MX=I
      CALL OUTPUT
C
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
      CALL INTM
      I=I+1
        L=3
      DO 3 IS=1,NTWO
        L=L+1
    3 U(IS,I)=T(L)
      MX=I
    4 CALL OUTPUT
      RETURN
      END
$IBFTC PANDH    LIST
      SUBROUTINE PANDH
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1   ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
     2   ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3   ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4   ,MX,FORCTX(9),ATRUE,CSPEED
C
      LFLAG=2
      DO 1 I=1,2511
    1 T(I)=0.0
      T(3)=DELTA
      NEQ=(NT+3)*N22
        L=4 + N22 + NT
      T(L)=1.0
      DO 2 I=1,NP1
```

```
      L=L + N22 + 1
    2 T(L)=1.0
      I=1
      DO 12 IS=1,NT
   12 UPREV(IS)=U(IS,I)
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      NEQ3=NEQ+3
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
      CALL INTM
      I=I+1
      DO  3 IS=1,NT
    3 UPREV(IS)=U(IS,I)
    4 CONTINUE
C
      L=3
      DO 5 IS=1,N22
      L=L+1
    5 P(IS)=T(L)
C
      DO 6 J=1,NP2
      DO 6 IS=1,N22
      L=L+1
    6 H(IS,J)=T(L)
C
      RETURN
      END
$IBFTC LINEAR  LIST
      SUBROUTINE LINEAR
      COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
     1 ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
     2 ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
     3 ,TT(9),U(18,401),P(19),H(19,10),C(10)
     4 ,MX,FORCTX(9),ATRUE,CSPEED
      DIMENSION AM(50,50),BV(50),IPIVOT(50),PIVOT(50),INDEX(50,2)
C
      DO 2 I=1,NT
      II=NT+I
      DO 1 J=1,NP2
    1 AM(I,J)=H(II,J)
    2 BV(I)=FORCT(I)/TENSN - P(II)
C
C
      DO 4 II=1,2
      I=NT+II
      DO 3 J=1,NP2
      AM(I,J)=0.0
      DO 3 IS=1,NT
    3 AM(I,J)=AM(I,J) + H(IS,J)*H(IS,I)
      BV(I)=0.0
      DO 4 IS=1,NT
    4 BV(I)=BV(I) + (UOBST(IS) - P(IS))*H(IS,I)
C
      DO 5 I=1,NP2
    5 PRINT9,(AM(I,J),J=1,NP2),BV(I)
    9 FORMAT(1X1P7E18.8)
C
      CALL MATINV(AM,NP2,BV,1,DETERM,PIVOT,INDEX,IPIVOT)
C
      DO 6 I=1,NP2
```

```
    6   C(I)=BV(I)
        A=C(NP1)
        B=C(NP2)
        PRINT9,(C(I),I=1,NP2)
        RETURN
        END
$IBFTC OUTPUT  LIST
        SUBROUTINE OUTPUT
        COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
       1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
       2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
       3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
       4  ,MX,FORCTX(9),ATRUE,CSPEED
C
        I=MX
        PRINT10,T(2)
   10   FORMAT(1H04X 3HX =, F10.6)
        PRINT11,(U(IS,I),IS=1,NT)
        PRINT11,(U(IS,I),IS=NP1,NTWO)
   11   FORMAT(1H01P7E18.6)
        RETURN
        END
$IBFTC DAUX    REF
        SUBROUTINE DAUX
        COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
       1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
       2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
       3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
       4  ,MX,FORCTX(9),ATRUE,CSPEED
        DIMENSION V(19)
        N21=NTWO + 1
        GO TO (100,200,200),LFLAG
C
C             NONLINEAR
C
 100      L=3
        DO 1 IS=1,N22
          L=L+1
    1   V(IS)=T(L)
          L=NEQ+3
        DO 2 IS=1,NT
          L=L+1
        NN=NT+IS
    2   T(L)=V(NN)
        DO 3 IS=1,NT
          L=L+1
          S=IS**2
        DENOM=V(N21) + V(N22)*T(2)
    3   T(L)=S*V(IS)/DENOM
          L=L+1
        T(L)=0.0
          L=L+1
        T(L)=0.0
        RETURN
C
C             LINEAR
C
 200      L=3
        DO 4 IS=1,N22
          L=L+1
```

```
    4   V(IS)=T(L)
          M=NEQ+3
        DO 5 IS=1,NT
          M=M+1
        NN=NT+IS
    5   T(M)=V(NN)
        DO 6 IS=1,NT
          M=M+1
          S=IS**2
        DENOM=A + B*T(2)
    6   T(M)=S*(V(IS)/DENOM - ( V(N21) + V(N22)*T(2))*UPREV(IS)/
       1  DENOM**2 + UPREV(IS)/DENOM)
          M=M+1
        T(M)=0.0
          M=M+1
        T(M)=0.0
C
        IF(LFLAG-3) 20,300,300
  300   RETURN
C
C              HOMOGENEOUS
C
   20   DO10 J=1,NP2
C
        DO 7 IS=1,N22
          L=L+1
    7   V(IS)=T(L)
C
        DO 8 IS=1,NT
          M=M+1
        NN=NT+IS
    8   T(M)=V(NN)
C
        DO 9 IS=1,NT
          M=M+1
          S=IS**2
        DENOM=A + B*T(2)
    9   T(M)=S*(V(IS)/DENOM - ((V(N21) + V(N22)*T(2)) *UPREV(IS))/
       1  DENOM**2)
          M=M+1
        T(M)=0.0
          M=M+1
   10   T(M)=0.0
        RETURN
        END
$IBFTC NEXT    REF
        SUBROUTINE NEXT
        COMMON T(2511),NT,RT(9),WT(9),UOBS1(9),FORCE(9),UOBST(9),FORCT(9)
       1  ,NPRNT,MPRNT,NP1,NTWO,KMAX,LFLAG,N22,NP2,B,BTRUE,NEQ
       2  ,DELTA,TENSN,A,UPREV(18),UOBSTX(9)
       3  ,TT(9),U(18,401),P(19),H(19,10),C(10)
       4  ,MX,FORCTX(9),ATRUE,CSPEED
C
        LFLAG=3
        NEQ=N22
        DO 1 I=1,2511
    1   T(I)=0.0
        T(3)=DELTA
          L=NT+3
        DO 2 I=1,NP2
```

```
      J=NT+1
      U(J,1)=C(I)
        L=L+1
    2 T(L =C(I)
        I=1
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      MX=1
      CALL OUTPUT
C
      DO 4 M1=1,MPRNT
      DO 3 M2=1,NPRNT
      CALL INTM
        I=I+1
        L=3
      DO 3 IS=1,NTWO
        L=L+1
    3 U(IS,I)=T(L)
      MX=I
    4 CALL OUTPUT
      RETURN
      END
```

APPENDIX G

PROGRAMS FOR WAVE PROPAGATION: MEASUREMENTS OF STEADY STATES

PROGRAM G.1. PRODUCTION OF REFLECTION COEFFICIENTS

The complete program is listed:

MAIN program

DAUX subroutine

The following library routine is required:

INTS/INTM

```
$JOB             2609,INDEX,K0160,3,100,100,C
$IBJOB INDEX     MAP
$IBFTC MAIN      LIST
CPHASE I
      COMMON T(399),N,NPRNT,MPRNT,DELTA,ZMAX,A,B,F(5),W(5),R(5),S(5),
     1  ETA.
C
C              INTEGRATE BACKWARDS FROM ONE TO ZERO
C
C              INPUT
C
    1 READ100,N,NPRNT,MPRNT
      PRINT90,N,NPRNT,MPRNT
      READ101,DELTA,ZMAX,A,B
      ETA=A
      PRINT91,DELTA,ZMAX,A,B,ETA
      READ101,(F(I),I=1,N)
      PRINT92,(F(I),I=1,N)
      TPI=2.0*3.1415927
      DO 2 I=1,N
    2 W(I)=TPI*F(I)
      PRINT93,(W(I),I=1,N)
C
C              INTEGRATE
C
      DO 3 I=1,399
    3 T(I)=0.0
      T(2)=1.0
      T(3)=DELTA
      RR=0.0
      DO 4 I=1,N
      R(I)=RR
    4 S(I)=0.0
      NEQ=2*N + 1
      J=NEQ   + 3
        L=3
      DO 40 I=1,N
        L=L+1
   40 T(L)=R(I)
      DO 41 I=1,N
        L=L+1
   41 T(L)=S(I)
      T(J)=ETA
C
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      CALL OUTPUT
C
      DO 8 M1=1,MPRNT
      DO 5 M2=1,NPRNT
    5 CALL INTM
        L=3
      DO 6 I=1,N
        L=L+1
    6 R(I)=T(L)
      DO 7 I=1,N
```

```
      L=L+1
    7 S(I)=T(L)
      L=L+1
      ETA=T(L)
    8 CALL OUTPUT
C
      GO TO 1
C
  100 FORMAT(6I12)
  101 FORMAT(6E12.8)
   90 FORMAT(1H14X 3HN =,I3,5X 7HNPRNT =,I3,5X 7HMPRNT =,I3)
   91 FORMAT(1H04X 7HDELTA =,E16.8,5X 6HZMAX =,E16.8/
     1            5X 3HA =,E16.8,5X3HB =,E16.8,5X 9HINDEX(0)=,E16.8)
   92 FORMAT(1H04X11HFREQUENCIES/5X5E20.8)
   93 FORMAT(1H04X19HANGULAR FREQUENCIES/5X5E20.8)
      END
$IBFTC DAUX    LIST
      SUBROUTINE DAUX
      COMMON T(399),N,NPRNT,MPRNT,DELTA,ZMAX,A,B,F(5),W(5),R(5),S(5),
     1  ETA
C
      DIMENSION RV(5),SV(5)
C
      L=3
      DO 1 I=1,N
      L=L+1
    1 RV(I)=T(L)
      DO 2 I=1,N
      L=L+1
    2 SV(I)=T(L)
      L=L+1
      ETA=T(L)
C
      ETAPR=2.0*B*(T(2)-1.0)
      ETA2=ETAPR/ETA
C
      DO 3 I=1,N
      L=L+1
    3 T(L)=0.5*ETA2 + 2.0*ETA*W(I)*SV(I)
     1  - 0.5*ETA2*(RV(I)**2 - SV(I)**2)
      DO 4 I=1,N
      L=L+1
    4 T(L)=-2.0*ETA*W(I)*RV(I) - ETA2*RV(I)*SV(I)
      L=L+1
    5 T(L)=ETAPR
      RETURN
      END
$IBFTC OUTPUT  LIST
      SUBROUTINE OUTPUT
      COMMON T(399),N,NPRNT,MPRNT,DELTA,ZMAX,A,B,F(5),W(5),R(5),S(5),
     1  ETA
C
      PRINT 10,T(2),ETA
   10 FORMAT(///1H04X3HZ =,F10.6,5X7HINDEX =,E16.8)
      PRINT11
```

```
      DO 1 I=1,N
      AMP=SQRT(R(I)**2 + S(I)**2)
      PHI=ATAN2(S(I),R(I))
    1 PRINT12,F(I),W(I),R(I),S(I),AMP,PHI
C
   11 FORMAT(1H013X11HFREQUENCY F, 7X13HANGULAR FREQ.,
     1  11X9HREAL PART, 11X9HIMAGINARY, 11X9HAMPLITUDE,15X5HPHASE)
   12 FORMAT(5X6E20.8)
      RETURN
      END
$ENTRY          MAIN
          3           10          100
      -.001          1.0          1.0          0.5
        1.0          2.0          3.0
$IBSYS          ENDJOB
```

PROGRAM G.2. DETERMINATION OF INDEX OF REFRACTION

The complete program is listed:

MAIN program

DAUX subroutine

OUTPUT subroutine

The following library routine is required:

INTS/INTM

```
$JOB            2609,INDEX2,K0160,5,0,100,P
$IBJOB          MAP
$IBFTC MAIN     REF
CPHASE2       INDEX OF REFRACTION  ETA(A,B)
C
      COMMON T(435),N,NPRNT,MPRNT,KMAX,DELTA,ZMAX,A,B,ETA,F(5),W(5),
     1  NTWO,N2P2,BOBS(10),LFLAG,N2P1,X(7,401),P(6,401),H(6,2,401),
     2  BVEC(2),AMAT(2,2),R(5),S(5),MX,NEQ
C
C             INPUT
C
    1 READ100,N,NPRNT,MPRNT,KMAX
      PRINT90,N,NPRNT,MPRNT,KMAX
      READ101,DELTA,ZMAX,A,B,ETA
      PRINT91,DELTA,ZMAX,A,B,ETA
      READ101,(F(I),I=1,N)
      PRINT92,(F(I),I=1,N)
      TPI=2.0*3.1415927
      DO 2 I=1,N
    2 W(I)=TPI*F(I)
      PRINT93,(W(I),I=1,N)
        NTWO=2*N
        N2P2=NTWO + 2
C             OBSERVATIONS
      READ101,(BOBS(I),I=1,NTWO)
      PRINT94,(BOBS(I),I=1,NTWO)
C
C             INITIAL APPROXIMATION
C
      K1=0
      PRINT95,K1
      PRINT97,A,B
        N2P1=NTWO + 1
        LFLAG=1
      DO 3 I=1,435
    3 T(I)=0.0
      T(2)=1.0
      T(3)=DELTA
      DO 4 I=1,NTWO
    4 X(I,1)=0.0
      X(N2P1,1)=ETA
        MX=1
      NEQ=N2P1
      J=NEQ+3
      T(J)=ETA
C
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
C
      DO 8 M1=1,MPRNT
      DO 5 M2=1,NPRNT
      CALL INTM
        MX=MX+1
      M=MX
        L=3
      DO 6 I=1,NTWO
```

```
        L=L+1
   6  X(I,M)=T(L)
        L=L+1
        I=N2P1
      ETA=T(L)
   5  X(I,M)=ETA
   8  CALL OUTPUT
C
C
C               SUCCESSIVE APPROXIMATIONS
C
      DO 25 K1=1,KMAX
      PRINT95,K1
C
C               PARTICULAR AND HOMOGENEOUS SOLUTIONS
C
        LFLAG=2
        M=1
      MX=M
      DO 9 I=1,NTWO
      P(I,1)=0.0
      DO 9 J=1,2
   9  H(I,J,1)=0.0
C
      NEQ=3*N2P2
      DO 10 I=1,435
  10  T(I)=0.0
      T(2)=1.0
      T(3)=DELTA
        L=3
      DO 11 I=1,N2P2
        L=L+1
  11  T(L)=0.0
      DO 12 J=1,2
      DO 12 I=1,N2P2
        L=L+1
  12  T(L)=0.0
        I=3 + N2P2 + N2P1
      T(I)=1.0
        J=3 + NEQ
      T(J)=1.0
C
      DO 13 I=1,N
      R(I)=X(I,1)
        J=I+N
  13  S(I)=X(J,1)
      ETA=X(N2P1,1)
C
      CALL INTS(T,NEQ,2,0,0,0,0,0,0)
      L3=NEQ+3
C
      DO 18 M1=1,MPRNT
      DO 17 M2=1,NPRNT
      CALL INTM
        M=M+1
```

```
      MX=M
        L=3
      DO 14 I=1,NTWO
        L=L+1
   14 P(I,M)=T(L)
        L=L+2
      DO 151J=1,2
      DO 15 I=1,NTWO
        L=L+1
   15 H(I,J,M)=T(L)
  151   L=L+2
      DO 16 I=1,N
      R(I)=X(I,M)
        J=N+I
   16 S(I)=X(J,M)
   17 ETA=X(N2P1,M)
   18 CONTINUE
C
C           BOUNDARY CONDITIONS DETERMINE NEW A, B
C
      DO 20 I=1,2
      BVEC(I)=0.0
      DO 19 K=1,NTWO
   19 BVEC(I)=BVEC(I) + H(K,I,M)*(BCBS(K)-P(K,M))
      DO 20 J=1,2
      AMAT(I,J)=0.0
      DO 20 K=1,NTWO
   20 AMAT(I,J)=AMAT(I,J) + H(K,I,M)*H(K,J,M)
C
      D=AMAT(1,1)*AMAT(2,2) - AMAT(1,2)*AMAT(2,1)
      A=(BVEC(1)*AMAT(2,2) - BVEC(2)*AMAT(1,2)) / D
      B=(BVEC(2)*AMAT(1,1) - BVEC(1)*AMAT(2,1)) / D
      PRINT97,A,B
C
C           NEW APPROXIMATION
C
        M=1
      MX=M
      T(2)=1.0
      X(N2P1,M)=A
C
      DO 22 M1=1,MPRNT
      DO 21 M2=1,NPRNT
        M=M+1
      MX=M
      T(2)=T(2)+DELTA
      X(N2P1,M)=A + B*(T(2)-1.0)**2
      ETA=X(N2P1,M)
      DO 21 I=1,NTWO
   21 X(I,M)=P(I,M) + A*H(I,1,M) + B*H(I,2,M)
   22 CALL OUTPUT
   25 CONTINUE
      GO TO 1
C
  100 FORMAT(6I12)
```

```
  101   FORMAT(6E12.8)
   90   FORMAT(1H06I20)
   91   FORMAT(1H06E20.8)
   92   FORMAT(1H06E20.8)
   93   FORMAT(1H08HOMEGA(I)/1X6E20.8)
   94   FORMAT(1H04X12HOBSERVATIONS /(5X5E20.8))
   95   FORMAT(1H14X13HAPPROXIMATION, I3)
   96   FORMAT(1H0F10.4/(5X5E20.8))
   97   FORMAT(1H04X 3HA =,E16.8, 5X3HB =,E16.8)
        END
$IBFTC DAUX     REF
        SUBROUTINE DAUX
        DIMENSION RV(5),SV(5),RPREV(5),SPREV(5),FUNR(5),FUNS(5)
        COMMON T(435),N,NPRNT,MPRNT,KMAX,DELTA,ZMAX,A,B,ETA,F(5),W(5),
       1 NTWO,N2P2,BOBS(10),LFLAG,N2P1,X(7,401),P(6,401),H(6,2,401),
       2 BVEC(2),AMAT(2,2),R(5),S(5),MX,NEQ
C
        GO TO (10,20),LFLAG
C
C              NONLINEAR
C
   10     L=3
        DO 1 I=1,N
          L=L+1
    1   RV(I)=T(L)
        DO 2 I=1,N
          L=L+1
    2   SV(I)=T(L)
          L=L+1
        ETA=T(L)
        ETAPRI=2.0*B*(T(2)-1.0)
        PR=ETAPRI/ETA
C
        DO 3 I=1,N
          L=L+1
    3   T(L)=0.5*PR + 2.0*W(I)*SV(I)*ETA - PR*(RV(I)**2-SV(I)**2)*0.5
        DO 4 I=1,N
          L=L+1
    4   T(L)=-2.0*ETA*W(I)*RV(I) - PR*RV(I)*SV(I)
          L=L+1
        T(L)=ETAPRI
        RETURN
C
C              LINEAR
C
   20   ETA=A + B*(T(2)-1.0)**2
        ETAPR=2.0*B*(T(2)-1.0)
        PR=ETAPR/ETA
        DNDA=1.0
        DNDB=(T(2)-1.0)**2
        DNPNDA=-ETAPR/ETA**2
        DNPNDB=2.0*(T(2)-1.0)/ETA -(ETAPR*(T(2)-1.0)**2)/ETA**2
C
CPARTICULAR
          L=3
```

```
      DO 5 I=1,N
        L=L+1
    5 RV(I)=T(L)
      DO 6 I=1,N
        L=L+1
    6 SV(I)=T(L)
        L=L+1
      ANEW=T(L)
        L=L+1
      BNEW=T(L)
      DO 7 I=1,N
      RPREV(I)=R(I)
    7 SPREV(I)=S(I)
      APREV=A
      BPREV=B
      DO 8 I=1,N
      FUNR(I)=0.5*PR + 2.0*ETA*W(I)*S(I) - (R(I)**2-S(I)**2)*PR*0.5
    8 FUNS(I)=-2.0*ETA*W(I)*R(I) - R(I)*S(I)*PR
      IFLAG=0
      M=NEQ+3
C
  100 IFLAG=IFLAG+1
      DO 101 I=1,N
        M=M+1
      T(M)=FUNR(I) + (RV(I)-RPREV(I))*(-R(I)*PR)
      T(M)=T(M) + (SV(I)-SPREV(I))*(2.0*ETA*W(I)  +  S(I)*PR)
      T(M)=T(M) + (ANEW-APREV)*(0.5*DNPNDA+2.0*W(I)*S(I)*DNDA
     1        - 0.5*(R(I)**2-S(I)**2)*DNPNDA)
  101 T(M)=T(M) + (BNEW-BPREV)*(0.5*DNPNDB+2.0*W(I)*S(I)*DNDB
     1        - 0.5*(R(I)**2-S(I)**2)*DNPNDB)
      DO 102 I=1,N
        M=M+1
      T(M)=FUNS(I) + (RV(I)-RPREV(I))*(-2.0*ETA*W(I)-S(I)*PR)
      T(M)=T(M) + (SV(I)-SPREV(I))*(-R(I)*PR)
      T(M)=T(M) + (ANEW-APREV)*(-2.0*W(I)*R(I)*DNDA-R(I)*S(I)*DNPNDA)
  102 T(M)=T(M) + (BNEW-BPREV)*(-2.0*W(I)*R(I)*DNDB-R(I)*S(I)*DNPNDB)
        M=M+1
      T(M)=0.0
        M=M+1
      T(M)=0.0
C
      IF(IFLAG-1)50,50,201
C
CHOMOGENEOUS
   50 DO 201 J=1,2
      DO 51 I=1,N
        L=L+1
   51 RV(I)=T(L)
      DO 52 I=1,N
        L=L+1
   52 SV(I)=T(L)
        L=L+1
      ANEW=T(L)
        L=L+1
      BNEW=T(L)
```

```
      DO 53 I=1,N
      RPREV(I)=0.0
      SPREV(I)=0.0
      FUNR(I)=0.0
   53 FUNS(I)=0.0
      APREV=0.0
      BPREV=0.0
      GO TO 100
  201 CONTINUE
      RETURN
      END
$IBFTC OUTPUT  REF
      SUBROUTINE OUTPUT
      COMMON T(435),N,NPRNT,MPRNT,KMAX,DELTA,ZMAX,A,B,ETA,F(5),W(5),
     1 NTWO,N2P2,BOBS(10),LFLAG,N2P1,X(7,401),P(6,401),H(6,2,401),
     2 BVEC(2),AMAT(2,2),R(5),S(5),MX,NEQ
C
      PRINT92
      DO 4 I=1,N
      R(I)=X(I,MX)
        J=I+N
    4 S(I)=X(J,MX)
      IF(MX-1)1,1,2
    1 PRINT91
    2 DO 3 I=1,N
      AMP=SQRT(R(I)**2+S(I)**2)
      PHI=ATAN2(S(I),R(I))
    3 PRINT93,T(2),X(N2P1,MX),I,R(I),S(I),AMP,PHI
C
   90 FORMAT(1H04X3HA =,E18.8,5X3HB =,E18.8)
   91 FORMAT(1HC10X1HX,6X8HINDEX(X),5X1HI,11X9HREAL PART,
     1         11X9HIMAGINARY,11X9HAMPLITUDE,15X5HPHASE//)
   93 FORMAT( F12.4,F14.6,I6,4E20.8)
   92 FORMAT(1H0)
      RETURN
      END
$ENTRY          MAIN
          3            20            20            5
      -.0025           1.0           1.1           0.4           1.1
        1.0            2.0           3.0
+.132178E-02+.323131E-03-.368550E-03+.148430E-01+.954147E-02+.589762E-02
```

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE COMPUTATIONAL ASPECTS OF INVERSE PROBLEMS IN ANALYTICAL MECHANICS, TRANSPORT THEORY, AND WAVE PROPAGATION	
4. AUTHOR(S) (Last name, first name, initial) Hagiwada, Harriet	

5. REPORT DATE August 1965	6a. TOTAL NO. OF PAGES 314	6b. NO. OF REFS. 194
7. CONTRACT or GRANT NO. SD-79	8. ORIGINATOR'S REPORT NO. RM-4549-ARPA	

9a. AVAILABILITY/LIMITATION NOTICES	9b. SPONSORING AGENCY Advanced Research Projects Agency

10. ABSTRACT

An investigation of inverse problems as basic problems in science, in which physical systems are to be identified on the basis of experimental observations. These problems are especially important in astrophysics and astronomy, for their objects of investigation are frequently not observable in a direct fashion. Solar and stellar structure, for example, is estimated from the study of spectra, while the structure of a planetary atmosphere may be deduced from measurements of reflected sunlight. This Memorandum shows that a wide class of inverse problems may now be solved with high-speed computers and modern computational techniques.

11. KEY WORDS

Celestial mechanics
Models
Mathematics
Astrophysics
Wave propagation
Inverse problems
Computers
Transport theory
FORTRAN